MULTIPLE UNMANNED VEHICLES OPERATIONS IN CONFINED AREAS

ZHANG QIAN

(B.Eng, Harbin Institute of Technology)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF MECHANICAL ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2015

DECLARATION

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

Zhang Qian

Zhang Qian

13 August 2015

Acknowledgements

First of all, I would like to express my sincerest gratitude to my supervisor, Assoc. Prof. Gerard Leng Siew Bing, for his continuous guidance and encouragement during my studies. He always gives me invaluable advice and shows me the direction, every time I feel confused. I am grateful for his support and patience over the years.

I would also like to present my gratitude to my fellow colleague, Vengatesan Govindaraju, for his discussion on research, encouragement and concern over the past years. I also wish to thank all my dear friends and all the staff in Dynamics Lab for their help and pleasant memories. I am also thankful to my friends, who always accompany me and give me confidence all the time.

I want to gratefully acknowledge China Scholarship Council (CSC) and the embassy of China for the financial support during my PhD study. I am truly thankful to National University of Singapore for the environment and resources provided.

Last but not least, I would like to deeply thank my parents for their consistent understanding and encouragement. They give me unconditional love and all-around support.

Table of Contents

Summary

The thesis aims to derive the time to first collision of multiple unmanned air/land/surface vehicles (UxVs) operating in a confined area. Here a collision is defined as two UxVs coming within a critical distance of each other. The effect of different vehicle and collision avoidance models are studied using the concept of a mean-free path inspired by molecular dynamics.

The time to first collision is derived for two cases of UxVs operating in a confined area. For the first case, the vehicles move with constant speeds with zero turn radius but have blind spots in detecting obstacles. The collision avoidance method is to turn 90° away from another oncoming vehicle. An expression for the time to first collision is derived as a function of the number of UxVs, the UxV speed and the sensor field of view (FOV) for a given operational area and vehicle size. The predicted time to first collision was verified by Monte-Carlo simulation. Furthermore, the theory indicates the existence of a critical time, above which collision is deemed to occur instantly. This critical time provides an estimate of the maximum number of UxVs that can safely operate in a given area.

In the second case, Dubins' vehicles were considered i.e. nonholonomic vehicles with constant speed and finite turn radius. The velocity obstacle method is used for collision avoidance. The time to first collision is derived in a similar manner and is now a function of the number of vehicles, speed as well as the vehicle's lateral acceleration. The theory agreed with the

Monte Carlo simulations and the critical number of UxVs that can operate safely increases with decreasing finite turn radius. The results provide useful guidelines for the safe operations of UxV in confined areas and the method may be applied to other vehicle models and collision avoidance methods.

List of Tables

List of Figures

List of Symbols

d	diameter of gas molecules/radius of virtual circles of gas molecules
Δt	time interval
ρ	density of gas molecules or UxVs
v, $\overline{v_{rel}}$	average relative speed between molecules or UxVs
$\bar{v}$	absolute speed of molecules or UxVs
λ	length of mean free path
C	a constant
r	radius of UxVs
A	rectangle area swept out by UxV
m, m_1	the number of vehicles that the reference vehicle will encounter in unit time
T	average time between two successive collisions of a UxV
R	radius of UxVs
$R_S{}'$	sensor range of UxVs
φ	an angle subtended by sensing region
R_S	radius of sensor range
α	the angle subtended by the connecting line between two centres of vehicles and sensing boundary
θ	the angle subtended by the connecting line between two centres of vehicles and x axis
$\overrightarrow{v_C}$	relative velocity component along the line of two centres of UxVs

P	probability of collision once two UxVs encounter
S	area of operation area
n	total number of UxVs in the confined area
$r(R, R_S)$	effective vehicle radius
C_1	the constant to convert average absolute velocity to average relative velocity
m_2	modification term
C_S	constant in the modification term
n_0	the number of collisions in unit time
T	the time to first collision in the confined area
C	constant in the formula
M, $\widetilde{M}$, M_1, M_2	substitution terms
n_{cr}	critical number of UxVs
T_{cr}	critical time to first time
x, y	position coordinates
dt	time step in simulation
${R_d}^2$	coefficient of determination
a, b, c	the equivalent constants when some variables are fixed
P_A, P_B	positions of the centres of vehicles a and b
P	position of a vehicle
v_A, v_B	velocities of vehicles a and b
v_{AB}	relative velocity between a and b
$r_A(R_a)$, $r_B(R_b)$	radii of vehicles a and b

r_{AB}	equivalent radius of a and b
$\lambda(P, v)$	the ray that start from P along the direction of v
VO_A^B	velocity obstacle of b relative to a
ω	angular speed
a	the maximum acceleration of a UxV
R_{ob}	radius of UxV
R_{ab}	equivalent safety distance
d_L	length of rectilinear motion
d_C	length of turning motion
R_{roc}	radius of curvature
N_r	the number of obstacle vehicles within the rectangular area
P_{cone}	the probability that the direction of relative velocity is within the cone
d_{AB}	distance between a and b
α_{cone}	half angle of the cone
P_{mean}	mean probability
$\bar{y}$	average value of function $y(x)$
β	certain angle turned by UxV
d_{cyce}	length of a cycle
N_{cycle}	the number of cycles in unit time
S_C	the area swept out by UxV while turning
$N_{collision}$	the number of collisions in a cycle
N	the number of collisions in unit time

$\bar{v}_r$	average relative velocity
$C(S, R_{ob})$	a term to make up the effect of boundaries
$f(x)$	the integral term in the formula
$T_n(x)$	polynomial of the nth order Taylor expansion
$R_n(x)$	remainder of the nth order Taylor expansion
$L_{\Delta t}$	the length of path during Δt
$N_{\Delta t}$	the number of collisions during Δt

Chapter 1. Introduction

Autonomous unmanned air/land/surface vehicles (UxVs) are playing important roles in many applications, due to their advantages over piloted vehicles. Particularly, multiple unmanned vehicle system, in which the vehicles can conduct a common task by cooperation without requirement of human control and supervision, has attracted much attention from researchers. UxVs can be used in dangerous or inconvenient environments where it is hard for humans to access or operate. Usually, sensors are mounted in vehicles, so that information about the behavior of the vehicle and the situation around it can be transferred to the operator, who is far from the workspace. This also provides the possibility for UxVs to be applied in military missions. In addition, a task may be completed by multiple low-cost cooperative UxVs more quickly and efficiently than a single UxV.

Task planning is a prerequisite for the deployment of multiple UxVs. In order to ensure the safety of the UxVs, a careful choice of parameters such as the vehicle density and allowable speed is necessary. Therefore, the objective of this thesis is to explore the time to first collision for multiple UxVs operating in a confined area. Different vehicle models and collision avoidance techniques will be discussed in the following chapters.

1.1 Background

1.1.1 Introduction to Multi-vehicle Systems

With the rapid development of sensors, control system, computer science and robotics, multi-vehicle cooperative system has become a research topic of much interest [1-3]. A multi-vehicle system is defined as a system of multiple dynamic entities that share information or tasks to accomplish a common, though perhaps not singular, objective. Multiple vehicles are more effective than a single vehicle robot in many tasks, for example localization and mapping. They can complete cooperative works which cannot be done by single vehicle, for example robo-soccer as shown in Figure 1.1. It has been observed that multi-vehicle systems can also accomplish tasks with less cost compared to a single vehicle with full capabilities. Therefore, a lot of research effort has focused on studying multi-vehicle system, and some challenges remain. The most basic problem in deploying multiple vehicles is to avoid collision among the vehicles. Some related researches involve communication, coordination, path planning and obstacle avoidance.

The vehicles in multi-vehicle system need to interact and cooperate with each other to conduct tasks, so the communication between them is quite important. There are many different ways of communication and decision making. In addition, path planning with collision avoidance is one of the most important issues for UxVs. The path can be scheduled according to the mechanics and dynamic constraints of the UxVs, as well as the environment in which the UxVs are maneuvered. There are many path planning and collision

avoidance methods, for example the potential field method. UxVs can be categorized into holonomic vehicles and nonholonomic vehicles. Our studies are conducted using nonholonomic vehicles, and two kinds of nonholonomic vehicles are applied.

Figure 1.1 Cooperative multi-UxV system [4]

1.1.2 Task Planning in Confined Area

Many issues, for example communication and localization, must be considered for a cooperative system of UxVs. Among all the issues on cooperative UxVs, the assurance of safety is the most basic one, and the maneuver of vehicles must be decided beforehand to avoid collision. While operating in the workspace, the UxVs have to avoid collision not only with each other, but also with the obstacles around them. For vehicles moving in a confined area, the

boundaries of the workspace also need to be avoided. Therefore, task planning before operating is extremely important, for example how many vehicles should work together at one time, and what the speed of them should be. The method of collision avoidance should also be considered to ensure the safety of vehicles. Collision checking is the first step to avoiding collision. This requires the definition of collision and the principle of checking collision. Consequently, the way to calculate the time to first collision for multiple vehicles will be studied in this research, so that we can know how the parameters of the mulit-UxV system affect collision probability. This study will make it possible to do task planning for some specific systems.

1.1.3 Simulation Tools

In this research, the mathematical simulation software MATLAB (matrix laboratory) is used to verify the theories that are developed and to find the parameters in the formula in some specific cases. MATLAB is a high level programming language, but can interface with the programs that are written in other languages, such as C, C++, Java. The algorithms can be developed faster than that developed by traditional languages, because low-level administrative tasks are not needed. The functions in MATLAB can also be integrated with other applications and languages. MATLAB was designed for numerical computing primarily, but now it is a powerful tool in solving mathematical problems and application development. Large amount of functions are provided to deal with problems like differential equations, Fourier analysis, filtering, integration and so on. The user interface is shown in Figure 1.2.

Many tools in MATLAB make it possible to develop algorithms

efficiently, including Command Window, MATLAB Editor, Code Analyzer and MATLAB Profiler. Processor-optimized libraries are used to execute matrix and vector computations faster. Besides, just-in-time (JIT) compilation technology is used to accelerate the speed than low-level programming languages.

An advantage of MATLAB is the graphical user interface, including GUIDE (GUI development environment). Simulation graphics can be seen and analyzed explicitly. In addition, many toolboxes are provided in MATLAB. The functions in toolboxes can perform some specific calculations conveniently. The Curve Fitting toolbox will be used in our research to validate the theory developed.

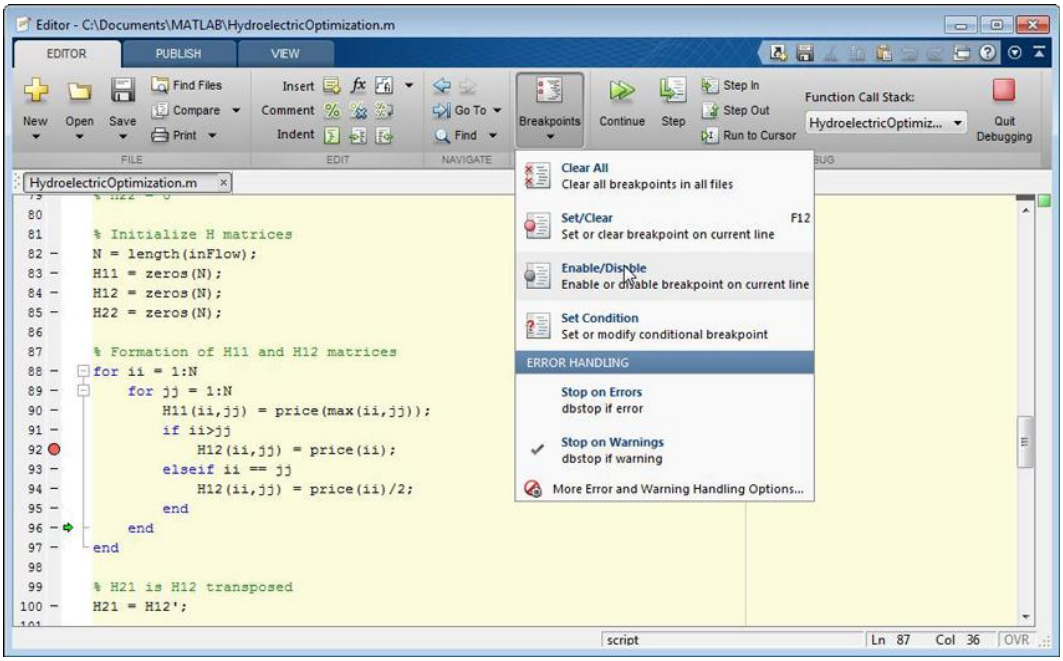


Figure 1.2 User interface of MATLAB [5]

1.2 Scope and Objectives

Multi-UxV system is still a challenging topic due to the communication between vehicles, path planning, etc, while the research in this thesis focuses on task planning of the vehicles. Safety is a significant premise of completing a task for multiple vehicles, so we always try to keep the probability of collision low. Therefore, before the beginning of any multi-UxV task, the planning on multi-UxV system is necessary to make sure a smooth operation. The planning includes the number of vehicles, range of sensors and so on. Different models of vehicles have different characteristics and motion pattern, so the planning should be made based on the scenario.

As introduced in section 1.1, many studies have been conducted on multi-UxV systems, including communication, control of motion, path planning, etc. All of the studies contribute to the development of effective and efficient multi-UxV system. Most of these studies focused on a specific problem to be solved, which cannot work well to other cases, and the studies rarely analyze the probability or conditions for collision to occur. Therefore, we would like to find a method derive the time to first collision in a confined area. In this research, we consider the case where the vehicles move freely with the same motion pattern. The time to first collision is studied in terms of different vehicle models and collision avoidance techniques. We study the effect of factors such as the number of vehicles within a confined area, vehicle speed and sensor range on the probability of collision and hence the time to first collision.

The objective of this research is to find the time to first collision within a confined area with respect to some different vehicle models and motion patterns. First, it is found that the motion for vehicles without collision avoidance in an open area is similar to that of gas molecules, so the derivation of Mean Free Path from molecular dynamics is used to derive the time to first collision. Different vehicles characteristics result in significant differences in the results, so the subsequent study focuses on the time to first collision in terms of two kinds of vehicle models and collision avoidance techniques. In the first part, a specific model of vehicle with constant speed and zero turn radius was proposed. The sensors on the vehicles have limited sensing range. Because of the existence of blind spots, collisions may happen. The influence of the factors to the time to first collision is identified and the effect of each factor is quantified. The second part is on the operation of Dubins' vehicles with finite turn radius. The collision avoidance technique used is based on the concept of Velocity Obstacle. The formula of the time to first collision is also derived. For both parts, the results were verified by Monte Carlo simulation.

A shorter time to first collision implies a higher probability of collision. Specifically, if we define a critical time, below which collisions are deemed to happen instantaneously, the relation among the parameters can be deduced. This relationship can then be used as a reference in planning UxVs operations within a confined area. The details of derivation will be described in the following chapters.

1.3 Contributions

In this thesis, the time to first collision in a confined area is derived, with respect to some different vehicle models and collision avoidance techniques. The main contribution of this research is as below:

- Some concepts and derivations in molecular dynamics are introduced in the study of multi-UxV system
- The formula for the time to first collision in a confined area is derived, with respect to two kinds of nonholonomic vehicles, and collision avoidance approaches. The influence of each factor is quantified by formula, so how the time to first collision varies with the factors can be easily analyzed
- The critical number of UxVs can be derived when critical time to first collision is specified. When the number of vehicles is below the critical number, the time to first collision can be expected to exceed the critical time
- The effect of the boundaries of workspace on the probability of collision is considered in this study
- Monte Carlo simulation is applied not only in verifying the theory developed, but also in approximating the constants in the formula

The method used in this research can be extended to other model of vehicles and motion patterns, based on the characteristics of the multi-UxV system and the requirement of the task. Therefore, the results can be used as a reference in task planning of multi-UxVs in a confined area.

1.4 Thesis Organization

The rest of this thesis is organized as below:

In Chapter 2, literature review is presented on previous works on Multi-robot system. Existing methods of path planning and collision avoidance are introduced, especially the concept of a Velocity Obstacle which is used in this research. Studies on Dubins' vehicle are also reviewed.

Chapter 3 discusses the time to first collision for vehicles with zero turn radius in a confined area. The time to first collision for vehicles without collision avoidance in an open area is first derived, referring to the derivation of mean free path in molecular dynamics. Next, the vehicle model is proposed, and the way of collision avoidance is specified. The formula of the time to first collision is obtained based on the model of the multi-UxV system. The critical number of vehicles is calculated, and Monte Carlo simulations were done to verify the formula that is developed. Subsequently, the study in Chapter 4 is on the time to first collision for Dubins' vehicles with non-zero turn radius in a confined area. Velocity obstacle is applied as the collision avoidance technique. The formula of the time to first collision is also derived in terms of this model, and the results are validated by Monte Carlo simulation. Besides, the constants in the formula are approximated by simulation.

Finally, conclusions of this thesis are presented and some recommendations for future work are discussed in Chapter 5.

Chapter 2. Literature Review

The system of multiple UxVs is a form of multi-robot system, so it is instructive to review recent research in the area of multi-robot system. The review in this chapter focuses on the challenges of multi-robot systems especially collision detection, avoidance and swarm robotics. In addition, studies on the characteristics of nonholonomic vehicles will be included.

2.1 Study Fields of Multi-Robot System

A multi-robot system consists of more than one autonomous mobile robot working together to complete a task, for example, search and rescue in a dangerous environment. In this section, we will review some important studies on multi-robot system communication [6, 7] [8] [9, 10] [11] [12] [13] [14] [15] [16], pattern formation [17], control [18], localization [19], especially collision detection and avoidance, which are most related to our studies.

2.1.1 Pattern Formation and Control Systems

Coordinated control of networked multi-robot systems is a problem which has attracted much attention. In particular, the pattern formation problem requires the robots to maintain a formation for task execution and involves coordination of multiple robots. Pattern formation is often categorized into centralized and decentralized pattern formation [20]. For centralized pattern formation, a central unit collects information from all robots and plans the motion of each robot. Instructions are then transmitted to the robots. A multi-

layer control scheme is proposed to deal with centralized UAV formation [21], which is an extension of previous work on nonlinear under-actuated controller. The controller can effectively coordinate the robots to move to specified positions and hence maintain a formation. Furthermore, this work can be extended to derive obstacle avoidance methods for the safe operations of multiple robots.

In a decentralized formation system, each robot can make its own decision and react regardless of the failure of other robots. Decentralized formation system is more flexible [22], so it will be applied in this thesis. A decentralized control algorithm for a swarm of robots based on the geometric approach is given in [23]. It combined the geometric approach and a simplified virtual physical mechanism for obstacle avoidance. This resulted in a robust and practical algorithm. The mechanism of swarm flocking phenomena is investigated in [24], and a distributed co-adaptive control algorithm is presented for a swarm robot system. The authors proved that the controller enabled all swarm members to converge to a common velocity using only local information, and the time to form the flock can be estimated. An analysis of Vicsek's model is introduced in [25], where multiple robots coordinate their motion by simple local nearest neighbour rules.

Motion constraint is also a fundamental issue in the control of robots. Such constraints may arise from the kinematics of the driving mechanisms of robots, e.g. rolling constraint, or conservation of angular momentum. For mobile robots without slipping, there is always a constraint on the velocity of the system which cannot be integrated into position constraints, which is

called nonholonomic constraint [26]. Nonholonomic constraint limits the freedom of motion of robot. Dynamic feedback linearization and small-gain methods may be applied to solve the problem for distributed control [27].

Many studies are conducted on fault tolerance of a multi-robot system [28]. In [29], a unified and distributed formation control architecture is proposed. It allows arbitrary number of robots to operate. The position and orientation of virtual centre can vary with time. It is a robust system that can tolerate the failure of some robots. Some vehicles may fail suddenly, and it is necessary to distinguish it from others. Souissi et al. [30] propose an approach to deal with this problem. It considers the case where some of the robots in the system may possibly fail by crashing. The algorithm ensures that the crash of faulty robots does not bring the formation to a permanent stop, and that the correct robots are thus eventually allowed to reorganize and continue moving together. The control of any formation shape is studied in [31]. The formation shape can be modified online and the number of robots can be increased or decreased online. Consequently, the approaches of formation control become more practical and reliable as more and more studies conducted on this problem.

2.1.2 Mapping and Localization

Mapping and localization is a fundamental task for multi-robot system. Simultaneous Localization and Mapping (SLAM) [1, 32-35] is a basic principle in study of robotics. SLAM is an important technique because the robots can accomplish a task without knowing the environment in advance. Large amount of information should be collected to localize the robots, so

sensor fusion is necessary to manage the information from different sensors. For example, Zhang et al. [36] propose an algorithm that combines sonar and laser sensor to complete SLAM. There are also many vision-based SLAM studies as well, which are efficient and effective in outdoor environments, such as MonoSLAM [37], FrameSLAM [38], Mni-SLAM [39]. A low-cost vision based SLAM approach [40] is proposed using lightweight sensors, and it can work in a wide range of conditions.

In terms of localization, pose estimation between vehicles can usually localize the vehicles. For example in [41], to tackle the problem of vehicle-to-vehicle (V2V) relative pose estimation that is essential for realizing cooperative localization, an indirect V2V relative pose estimation (InDV2VRPE) method is proposed, which overcomes the disadvantages of direct V2V relative pose estimation methods. Mapping and localization is still a field that remains challenging as the development of robots and complexity of environments. For the studies in this thesis, instead of global localization, where each UxV knows the positions of all other UxVs, the UxVs have front mounted sensors with limited range to determine the positions of UxVs in its local neighbourhood.

2.1.3 Collision Detection and Assessment

Safety is a significant consideration in autonomous operation of unmanned vehicles, so reliable methods of collision detection and avoidance are highly desired. The geometrical approach [42] and the probabilistic approach [43] are commonly used in detecting collision. However, in order to ensure the safety of vehicles, collision risk assessment is the first step. State propagation [44]

and model-based approaches [45] are sometimes used to predict collision. The probability of collision can be determined by the time to collision [46]. A stochastic model method is proposed in [46] to assess the collision risk. The collision risk is studied by switching the coefficients of the stochastic differential equation. Du Toit et al. [47] presents a probabilistic collision checking between uncertain configurations for two objects, which is referred to as collision chance constraints. In [48], a platform is developed to complete the process from collision detection to avoidance. The position of obstacle can be calculated and the threat of collision can be sent to an agent to manage the threat. Belkhouche et al. [49] propose a model of collision risk detection and assessment for autonomous air vehicles. As uncertainties always exist in the system, the collision conditions on both deterministic case and uncertain case are discussed. The formulation in the paper has obvious simplifications, since it is not necessary to know the information about speed and orientation explicitly. Collision avoidance activates when the probability of collision is beyond some specific threshold.

One of the most useful and well-known methods to detect and avoid obstacles is the Velocity Obstacle method [50], which will be applied in this study. The main idea of velocity obstacle is as below: A and B are two vehicles, and a set of relative velocities of a vehicle that will lead to collision are found to form a velocity obstacle. If the relative velocity of the vehicles at current time is within the velocity obstacle, the vehicles will collide with each other, assuming that they move with current velocities. Therefore, the vehicle needs to select a new velocity outside the velocity obstacle, so that collision will not happen. Recently, the concept of velocity obstacle has been extended to adapt

to some specific conditions, for example Loss of Communication Obstacle (LOCO) [51]. LOCO is proposed as an additional constraint to maintain team coherence. The velocity of a robot is selected from the set that avoid both LOCOs and Velocity Obstacles, so both coherence maintenance and collision avoidance can be fulfilled. It has also been extended to reciprocal velocity obstacles, taking into account the other moving entities to prevent oscillation [52]. Velocity-acceleration obstacle has also been extended to consider acceleration constraints[53]. Because of the efficiency of the Velocity Obstacle method, it is used in Chapter 4 to detect potential collisions, and the velocities of vehicles are changed to avoid collision.

2.1.4 Path Planning and Collision Avoidance Methods of UxVs

After assessing the collision risk, collision avoidance and path planning will be the next most important requirements for safe operation of multiple vehicles [28]. A detailed review on conflict modeling and resolution methods is found in [54]. The problem of collision avoidance has been thoroughly studied for one robot avoiding static or moving obstacles [55]. More attention is required for the more involved and less studied problem of multi-vehicle collision avoidance (or any decision-making entities)[56-60]. This problem has important applications in many areas, such as multi-vehicle navigation and coordination among swarms of robots.

The potential field approach is a widely used and long established method [61]. A modified potential field approach is introduced in [62]. In [62], the controllers are implemented on each robot, so a distributed leader-follow architecture is used. The information about the position of the leader or virtual

leader and the position of each robot is collected, so that the robots can track the leader. In order to avoid collision, when the obstacle enters the detection area of a robot, the position of the robot and the obstacles are transferred to the controller on the corresponding robot. The advantage of this method that is dramatically different from some other potential field approaches is that, the algorithm of this collision avoidance control is in real time and that the robots only need to detect obstacles in its neighbourhood by using a locally defined potential functions.

On the other hand, there are some weaknesses of such established methods, for example some conflicts cannot be solved by just changing velocities. So, more comprehensive methods are developed. In [63], conflicts are detected using an algorithm based on axis-aligned minimum bounding box. The detected conflicts are solved by a genetic algorithm. The overall minimum cost is calculated, and the trajectories of the robots are modified based on the cost function. The initial flight plan of each robot will be changed by adding intermediate waypoints. The solution of flight plan will maintain the velocities of robots.

In addition, many other approaches are proposed to deal with collision avoidance problem [64, 65]. An efficient and practical collision-avoidance mechanism is developed for multi-agent system in [66]. Some strategies of controlling the vehicles to avoid collision are also proposed, such as navigation functions' based methodology[67], prediction algorithm [68], Bernstein–Bézier curves [69], limit cycle method [70], optimality and learning [71].

Most of the studies on collision avoidance focus on the time interval just before collision. Path planning is another effective way to avoid collision, which is a global planning of the motion of vehicles. The main objective is to find an optimal path from the starting point to the end point, which avoids physical obstacles, threats and evadable zones, while satisfying the performance requirement of multi-robot systems. Common methods of path planning are: A* algorithm [72-74], genetic algorithm (GA), simulated annealing (SA), artificial neural networks (ANN), dynamic programming algorithm [75], particle swarm optimization (PSO), Linear Programming (LP) [76], and etc [77-79].

Therefore, the problem of avoiding collision and ensuring a safe operational environment has been well studied, but little work has been done to find the factors affecting the probability of collision. All the studies above focus on the effectiveness or efficiency of some specific collision avoidance methods, and the methods are improved gradually. Many typical approaches have been proposed, which are effective for collision avoidance. Some approaches will be used in this study to avoid collision, and the time to first collision under some specific conditions will be derived. In this thesis, we will explore how the properties of the vehicles and environment affect the probability of collision in a confined area.

2.2 Swarm Robotics

Our studies are conducted on a large number of relatively simple robots, and such robots that have intelligent behaviours are known as swarm robots [80-

82]. Therefore, design and analysis of swarm robotic system will be reviewed in this section. Swarm robotics is inspired by social insects, such as ants, bees, as social insects can behave robustly and flexibly. “Swarm robots” was first proposed as “cellular robots” in [83] to indicate a general type of cellular automaton. The author gave an introduction in detail on the development of “swarm” [84]. It can be seen that researchers express significant interest in swarm systems over recent decades. Targets searching [85] is a main domain of application of swarm robotic systems. As a large number of robots are distributed in the space, the region can be covered to search for the target. Dangerous works or tasks that require redundancy can also be done by swarm robotic system. Another advantage of swarm system is that the scalability of the system can be changed easily, so it can be applied to tasks that require scale-up or scale-down in real time [81]. Therefore, lots of studies focus on cooperative swarm systems for all kinds of applications [86-89].

2.2.1 Design of Swarm System

There is no a formal way to design an individual swarm robot so that the desired collective behaviour is generated. A common design approach is behaviour-based [82]. As an individual robot cannot plan its motion, the probability of the whole collective behaviour is always used to describe the system, so probabilistic finite state machines (PFSMs) is commonly applied in the design of swarm system [90]. The transition probability is a function of the parameters of system. The parameters can be fixed, so that the transition probability is also a constant. For example, a genetic aggregation behaviour is proposed in [91], and the parameters of the system and environment were

varied systematically. Another commonly used behaviour-based design method is virtual physics-based design, which is inspired from physics [82]. A very general framework was proposed in 2004 [92], which is called "physicomimetics" or "artificial physics". The robots react to virtual forces effectively, so this method is always described by this framework. Virtual physics-based design methods can be easily applied into the entire system without additional rules. Moreover, the properties of the system can be derived by physical theories and tools. Therefore, virtual physics-based method is often used in design.

In some other systems, the robots can generate their behaviours automatically without the help of centralized coordination. The main approach in designing swarm robotics is evolutionary robotics. Neural network is used in this method to predict the system, and the parameters are decided by evolutionary algorithm. However, limitations still exist in evolutionary robotics, and current evolution approaches for the design of swarm robotics are not adequate. Furthermore, the approaches are not capable enough to provide solutions in practical applications. Some efforts have been made to fill the gap. A novel approach to the automatic design of control software for swarm robotics, AutoMoDe [93], is proposed. Control software can be designed automatically to accomplish aggregation and foraging [94].

2.2.2 Behaviour Analysis

The collective behaviours are usually modelled at microscopic level or macroscopic level. The microscopic models can be either simple as point masses, or complex with dynamic models. The simulation of swarm robotic

system is very similar to that of common mobile robotics system. The greatest difference between them is that swarm robotic system needs to consider large number of robots. However, the majority of multi-robot simulators do not take the number of robots into account. A simulator is developed to deal with the problem of scalability of swarm robots [95]. Besides, the swarm robotic system which uses centralized algorithm is always not scalable in terms of computation cost [96]. For such systems, the decision on the number of robots that operate in the workspace is especially important. Therefore, in this thesis, we will explore how the number of the vehicles and the property of environment affect the probability of collision in a confined area.

Besides the model of individual robot, swarm robotic system can be modelled macroscopically. Many works uses rate equations to model a swarm robotic system [97], which can be used to describe the rate of the number of robots that in some specific state over the total number of robots.

2.3 Nonholonomic Vehicles

Nonholonomic constraint is a kind of non-integrable kinematic constraint. Most of robot vehicles are nonholonomic, and the vehicles used in this thesis are all nonholonomic vehicles. When the dimension of the space that is achievable by a robot is smaller than the dimension of the robot's configuration space, the robot has nonholonomic constraints [98]. Common vehicles have typical nonholonomic mechanisms. As the velocity of a vehicle is always tangent to the orientation of the vehicle, the dimension of the achievable velocities of a vehicle is less than the dimension of its

configuration space. The nonholonomic characteristics of vehicles have to be considered in study of mobile robots [99, 100].

Dubins' vehicle is a typical kind of nonholonomic vehicle, which has been widely studied. Dubins first proposed the vehicle model in 1957 [101]. The vehicle is restricted to a planar, and kinematic model is applied extensively in a wide range of studies, including path planning [102, 103], coverage problem [104], robust control [105], etc. Many studies on multiple Dubins' vehicles have been conducted. In [104], the coverage problem can be done by multiple vehicles with the worst-case traveling time. Efficient loitering patterns are proposed for multiple vehicles to solve disk-covering problem.

2.4 Conclusion

In this chapter, we made a review on recent researches of multi-robot systems, and especially swarm robotic system was discussed. Studies on nonholonomic vehicles are also introduced. It can be seen that multi-robot systems have played an important role in a variety of applications, and many challenges still remain in this area. As described in this chapter, safety and task planning are significant issues in multi-robot system. Therefore, the time to first collision of multiple nonholonomic UxVs with collision avoidance methods in a confined area will be studied in this thesis. The factors that affect the time to first collision will be derived and quantified. The critical number of vehicles will be specified to ensure that collision happens beyond the defined time, so that a plan of the whole system can be made before starting a task.

Chapter 3. Time to First Collision for Vehicles with Zero Turn Radius in a Confined Area

Sometimes we need first to decide on the parameters of a multi-UxV system, such as the vehicle density, the speed of vehicles that operate in an area, before the vehicles begin a task. The planning can decrease the probability of collision between vehicles, so that the vehicles can operate safely. We want to know the relation between the time to first collision among the vehicles and the parameters in different cases. First we will derive the time to first collision for vehicles operating in an open area without collision avoidance. However, in more practical situations, the operating area is always bounded, and the vehicles need to avoid collision with each other. Therefore, in the subsequent sections, we will derive the time to first collision for vehicles with zero turn radius in a confined area, and the vehicles have a simple collision avoidance method. The formula of the time to first collision will be derived, so that the effect of each factor can be quantified. A better knowledge of the factors that influence the time to first collision will help us design multi-agent systems with low collision probabilities. For example, when the speed of the UxVs is too large, collisions are invariably inevitable. A limited FOV with significant "blind-spots" can also result in frequent collisions. The time to first collision is related to the probability of collision. A shorter time to first collision implies a higher probability of collision. Specifically, a critical time is defined as the

desirable least time to first collision, below which collisions could be deemed to happen instantly. The formulae that are derived in this chapter will help find the proper number of vehicles, speed and sensing range while operating in a confined area according to different task requirements, and the model can be modified to find the critical values for a specific factor. In this chapter, a simple model of the vehicle will be proposed first, and then we will introduce the theory for calculating the expected time to first collision in a confined area.

3.1 Time to First Collision for Vehicles without Collision Avoidance in an Open Area

We first begin with a simple case where the vehicles are operating in an open area. The time to first collision in an open area will be derived based on the concept of Mean Free Path [106]. Assuming that infinite unmanned vehicles distribute uniformly in an open area, and move randomly without collision avoidance, it is observed that the motion of vehicles is very similar to that of molecules in an open area. Therefore, the time to first collision among the vehicles can be inspired by the derivation of mean free path in molecular dynamics.

In this section, we would like to introduce the details of the derivation of mean free path for molecules in an open area. We will also show the derivation of the time to first collision for circular unmanned vehicles in an open area, so the time to first collision can be adjusted by changing the parameters.

3.1.1 Introduction to Mean Free Path

3.1.1.1 Basic Principles in Physics

Mean free path is a concept from molecular dynamics. Gas molecules always collide with each other while moving, so they cannot move in a straight path for a long time. However, they move in a straight line between two successive collisions, and their directions of motion and speeds will be changed after an impact. The average distance that a molecule travels between two collisions is the mean free path for specific kind of molecules in certain circumstance, e.g. temperature, pressure.

In the following, the derivation of the mean free path of gas molecules will be introduced in detail, and the idea of our work in the following is inspired from mean free path. The theory will be useful in finding the probability of encounter between two vehicles.

We use a figure to show the condition of collisions, and how to determine a collision. We assume that the gas molecules are spheres of diameter d, and there is a virtual circle around the molecule, as shown in Figure 3.1. This figure shows the case when two molecules just collide. A collision will take place if the distance between the centres of two molecules is less than the diameter of a molecule d. This model can be equivalent to a molecule with radius d (the virtual circle in the figure) and a mass point. Once the mass point is within the virtual circle, we can say that two molecules collide.

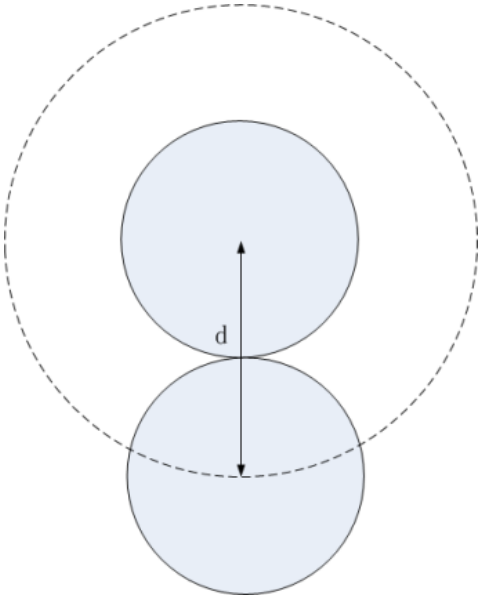


Figure 3.1 Critical condition of collision for two molecules. The solid circles show the vehicles with diameter d, and the dashed circle shows the virtual circle with radius d.

As stated above, a molecule can move along straight path between two successive collisions. When a molecule moves in the three-dimensional space along a straight path, it will sweep out a cylinder, and if any part of other molecules is within the cylinder, collision happens, as shown in Figure 3.2. Suppose that we select a reference molecule, in a small time interval Δt when the reference molecule moves along straight line, and it sweeps out a cylindrical region in space (i.e. the inner cylinder in Figure 3.2). We also assume that the reference molecule has a virtual circle as in Figure 3.1, so the virtual circle sweeps out the outer cylinder, and if the centres of other molecules are within the outer cylinder, the molecules will collide with the reference vehicle. The red dots in the outer cylinder in Figure 3.2 show the positions of the centres of other molecules. Here, we assume that only the reference molecule is moving, and all the others are static. Therefore, the reference molecule will collide with any molecule in the region bounded by

the outer cylinder.

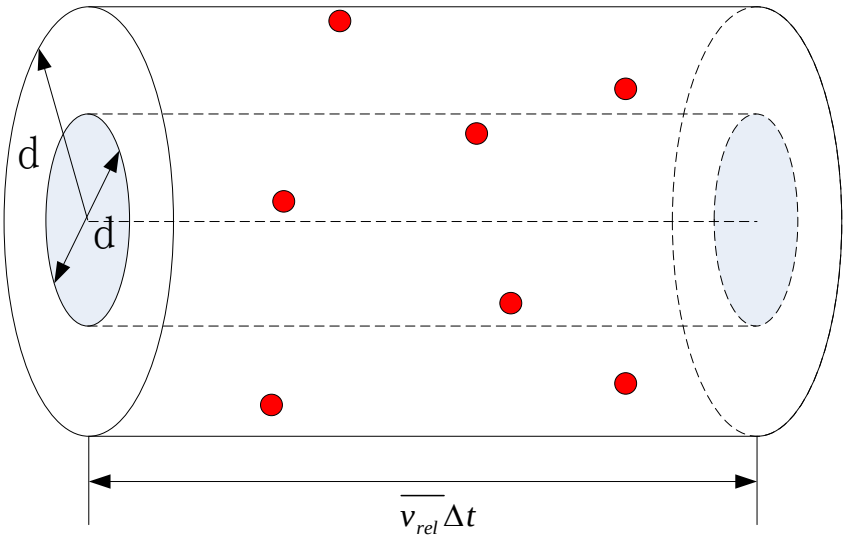


Figure 3.2 The dashed inner cylinder is swept out by the solid circle in Figure 3.1, while the solid outer cylinder is swept out by the virtual circle in Figure 3.1. The red dots represent the centres of molecules that are within the outer cylinder.

3.1.1.2 Calculation of Mean Free Path

According to the physical principle in last section, the formula of mean free path will be shown in this section. Assume that the number of molecules per unit volume is ρ, and the area of the cross section of the outer cylinder is πd^2 (see Figure 3.1). If v is the speed of the reference molecule, and the molecule moves in a time interval Δt, we can see from Figure 3.2 that the volume of the cylinder is $\pi d^2 v \Delta t$, so the number of molecules in the cylinder is $\rho \pi d^2 v \Delta t$. Because all the molecules within the cylinder will collide with the reference one, when $\Delta t = 1$, we can get that the number of collision in unit time n_0 is

$$n_0 = \rho \pi d^2 v \tag{3.1}$$

Because the mean free path is the average distance travelled by a moving molecule between successive collisions, we only need to find the number of

collisions in some time interval, and the mean free path can be calculated as the ratio of distance traveled by the molecule and the number of collision in this time interval. Therefore, mean free path can be derived by

$$\lambda = \frac{L_{\Delta t}}{N_{\Delta t}} \tag{3.2}$$

where $L_{\Delta t}$ is the length of path during Δt, and $N_{\Delta t}$ is the number of collisions during Δt. We assumed that only the reference molecule is moving, so the speed of the reference molecule v is equivalent to the relative speed to other molecules, if we consider the motion of other molecules. In reality, the rest of molecules also move in random speeds and directions. Therefore, we have to substitute the relative speed into the absolute speed of the reference molecule. The average relative speed can be found from the molecular speed distribution using the relation

$$\overline{v_{rel}} = C\bar{v} \tag{3.3}$$

where C is a constant, and $\bar{v}$ is the absolute speed. Relative speed should be used when calculating the number of collisions during the time interval Δt. However, other molecules have no effect on the path of the reference molecule, so we still use the absolute speed in calculating the length of path. Therefore, after substitution, the formula of mean free path is

$$\begin{aligned} \lambda &= \frac{L_{\Delta t}}{N_{\Delta t}} \\ &\approx \frac{\bar{v}\Delta t}{\pi d^2 \overline{v_{rel}} \Delta t \rho} \\ &= \frac{1}{C\pi d^2 \rho} \end{aligned} \tag{3.4}$$

The density of the molecules ρ is determined by the properties of the gas and environment, such as the diameter of molecule, temperature, pressure.

3.1.2 Time to First Collision Using the Mean Free Path

Now, we consider the motion of vehicles without collision avoidance in an open area. The vehicles are assumed to be circles. When the vehicles move freely in the space without collision avoidance, the motion of such vehicles is very similar to the motion of molecules. We only study the vehicles in two dimensions in this thesis, but we can still derive the number of collisions in unit time by applying the derivation of mean free path.

Instead of the cylinder swept out by a molecule in three dimension, the reference vehicle can only sweep out a rectangle when it moves in the operating area between two successive collisions, as shown in Figure 3.3. Let the radius of vehicle be r. The inner rectangle in Figure 3.3 is swept out by the vehicle, while the virtual circle with radius $2r$ sweeps out the outer rectangle. Similar to the derivation of mean free path, we let the vehicle move a small distance $\overline{v_{rel}}\Delta t$ in a time interval Δt, where $\overline{v_{rel}}$ is the average relative speed. The area of the outer rectangle is $4r\overline{v_{rel}}\Delta t$, which we concern in calculating the number of collisions. The vehicles in the rectangle are assumed to be mass points. In unit time ($\Delta t = 1$), the rectangular area is

$$A = 4 \cdot r \cdot \overline{v_{rel}} \tag{3.5}$$

If the density of vehicles is ρ, the number of vehicles in the rectangle in unit time can be obtained by multiplying the area by density. In other words, the number of vehicles that the reference vehicle will encounter in unit time is

$$m = \rho A = 4r\overline{v_{rel}}\rho \tag{3.6}$$

Now, we can derive the average time between two successive collisions, which is the reciprocal of the number of collisions in unit time

$$T = \frac{1}{m} = \frac{1}{4r\overline{v_{rel}}\rho} \tag{3.7}$$

The average time between two collisions also indicates the time to first collision from beginning in the operating area. After substituting absolute speed for relative speed, the formula for the time to first collision is

$$T = \frac{1}{4rC\bar{v}\rho} \tag{3.8}$$

where C is a constant.

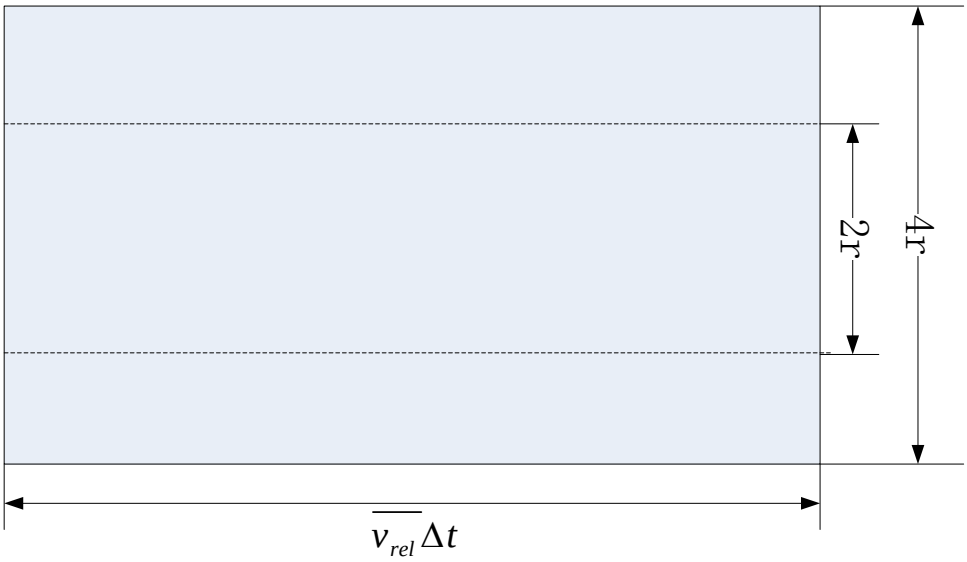


Figure 3.3 The inner rectangle is swept out by the reference vehicle, and the outer rectangle is swept out by virtual circle.

Consequently, formula Eq.(3.8) is the time to first collision for vehicles without collision avoidance in an open area. In this section, the concept and derivation of mean free path are introduced. Mean free path is a concept in

molecular dynamics, which means the average distance that a molecule travels between two successive collisions. We try to derive the time to first collision for multiple vehicles moving freely in an open area. In this case, the motion of vehicles is very similar to that of molecules, so we applied a part of derivation of mean free path into our study. As a result, the formula for the time to first collision in an open area is shown, which is determined by the size, speed and density of the vehicles.

In the next section, a more specific vehicle model with collision avoidance will be proposed, and the motion of vehicles in a confined area will be studied. The time to first collision will also be derived.

3.2 Model of Vehicle

In many studies, a safe distance is always defined to ensure the safety of the vehicles. As a commonly used definition, the safety area in this study is defined as a circle, the centre of which is at the origin of the vehicle, and the radius of the circle is the safe distance that is set. If an obstacle is within the safety distance from the vehicle, collision is deemed to have happened. Considering the safety distance, we assume the vehicle as a circle of radius R for simplicity (safety distance is included), corresponding to the inner circle in Figure 3.4. It is proposed that each vehicle can only perform two kinds of motions at any time: forward translation along its ‘y’ body axis and rotation about its ‘z’ body axis (pointing out of the plane). These two motions cannot be executed simultaneously. We suppose that a sensor is mounted in the front of the each vehicle (point G) and aligned with the y body axis. The sensor range

is $R_S{}'$ and its actual field of view (FOV) is indicated by the range AGB. Many kinds of sensors can generate a sensing range like this, for example sonar and infrared sensor [107-109]. However, for better understanding of the following derivation, sector AOB is used to indicate the FOV, i.e. the hatched region which subtends an angle φ at the origin of the body axes, point O. When the size of vehicle and the radius of sensing range are known, the relation between angle AGB and angle AOB can be easily derived geometrically. The angles can be converted if necessary. Therefore, this will not change the final results. The sensing area, which is in the shape of the difference of two sectors, is symmetrical about symmetry axis of the vehicle, which is also the direction of velocity when the vehicle is moving forward. In this study we define $|\overline{AF}| = R_S$ which simplifies the subsequent derivation (see Figure 3.4). The size of the sensing range is described by radius R_S and sensing angle φ.

If an obstacle or another vehicle is detected in the FOV, the detecting vehicle will rotate 90° away from the obstacle and continue moving forward. For example, if the obstacle vehicle is on the right hand side relative to the host vehicle, and it is detected, the host vehicle will rotate 90° to the left hand side to avoid the obstacle, as shown in Figure 3.5.

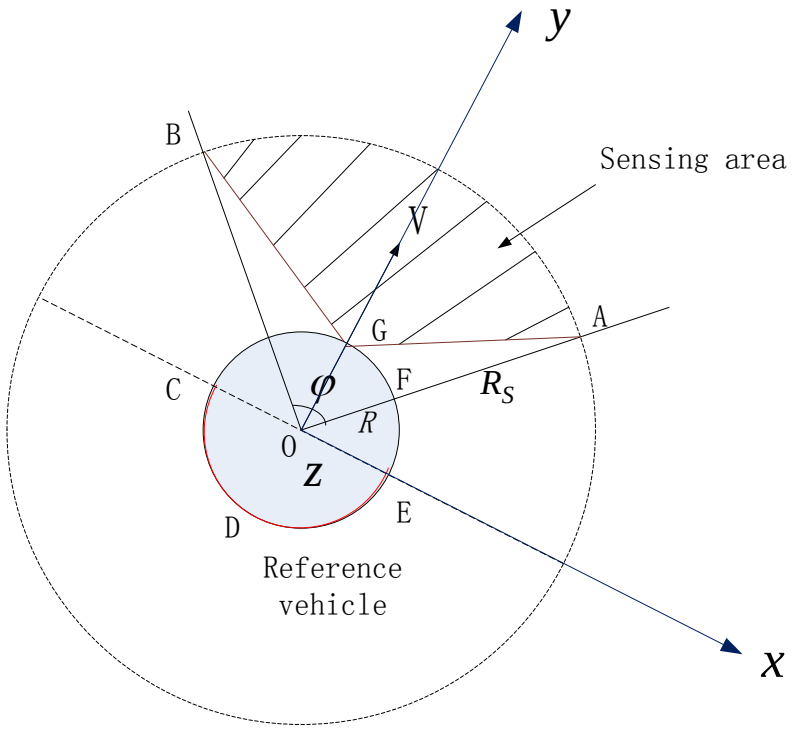


Figure 3.4 Model of vehicle is a disk centred at point O with a sensor mounted at point G. The shaded area is sensing area.

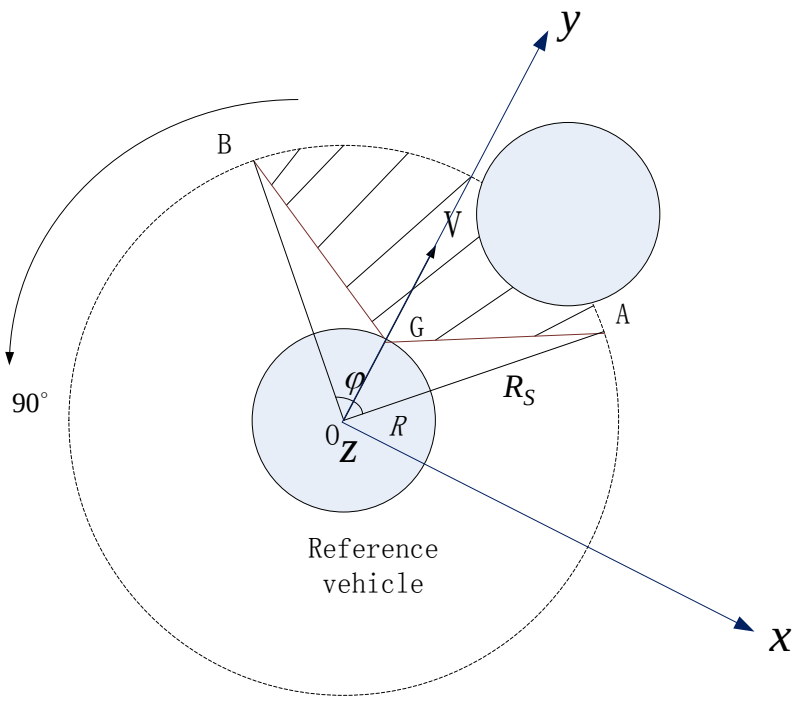


Figure 3.5 Vehicle rotates 90° to avoid the obstacle.

With the vehicle model defined above, we may now discuss the collision geometry. For the two vehicles shown in Figure 3.6, the obstacle vehicle is just beyond the FOV of the reference vehicle, so this is a critical case where the reference vehicle can just exactly detect the obstacle vehicle. Let $\overline{OO'}$ be the line that joins the centres of the two vehicles. This line subtends an angle θ with x body axis. Define the angle between $\overline{AO}$ and $\overline{OO'}$ as the intersection angle α. For a collision to occur, $\overline{OO'}$ must lie within the angle range θ, under which condition the reference vehicle cannot detect the obstacle. Since the vehicle cannot move backward, two vehicles will never collide with each other with the rear parts, the semicircle $\overline{CDE}$ shown in Figure 3.6. Besides, if two vehicles are at the state that is shown in Figure 3.6, it can be seen that the reference vehicle just right cannot detect the obstacle vehicle. While two vehicles moving closer along the line connecting two centres of them, they will also not collide, because in practical situation the sensing range will be smaller, and the host vehicle still cannot detect an obstacle. Therefore, blind spot exists when $\overline{OO'}$ is within the angle range θ, which means that if the centre of obstacle vehicle is within the sector area of angle θ and they have a relative speed which tends to make two vehicles closer, collision will occur. In other words, if the reference vehicle collides with an obstacle vehicle or the boundary of operating area, the collision point must be within the angle θ. Furthermore, one vehicle has two blind spots locating at left and right side respectively, so the field of view is of angle $\varphi + 2\alpha$.

The vehicles move according to the rules above in the operating area, and the motions will be discussed in following sections.

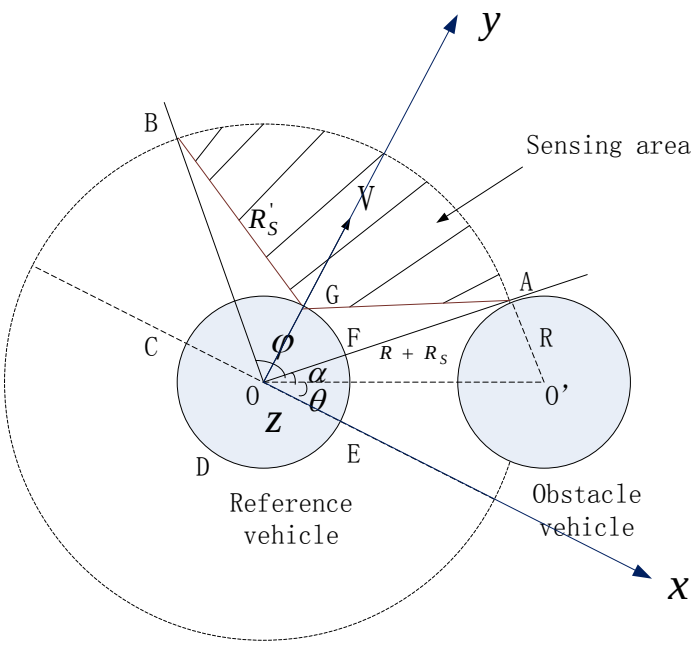


Figure 3.6 The critical case where the reference vehicle O just cannot detect the obstacle vehicle O'.

3.3 Derivation of formula for the Mean Time to First Collision

After the model and motion pattern of vehicles are introduced in previous section, the formula for the mean time to first collision will be derived in this section. First, the probability of collision for two vehicles will be found based on the geometrical relation and motion pattern of the vehicles. Furthermore, the number of collisions in unit time will be calculated, and shown in a function of the parameters of vehicles.

3.3.1 Probability of Collision for Two Vehicles

Unlike gas molecules, the presence of sensors may prevent collision to happen even if vehicles encounter each other, and we define the number of vehicles

that some specific vehicle will encounter in unit time as m. Before we derive m, the probability of collision for two vehicles will first be derived in this part when encounter happens. An encounter is defined as any two parts of two vehicles (including the sensing region) contact. Because the orientation of velocities and relative position of two vehicles are random when they encounter, which can be seen from the derivation results in the following, we need to find the probability that two vehicles encounter. There are two necessary conditions on relative orientation of velocities and relative position of two vehicles respectively. Collision will happen if two conditions are satisfied on the premise that they encounter. The relative position can be revealed by the contacting point, and point A in Figure 3.7 is an example. The following analysis will present the two conditions respectively:

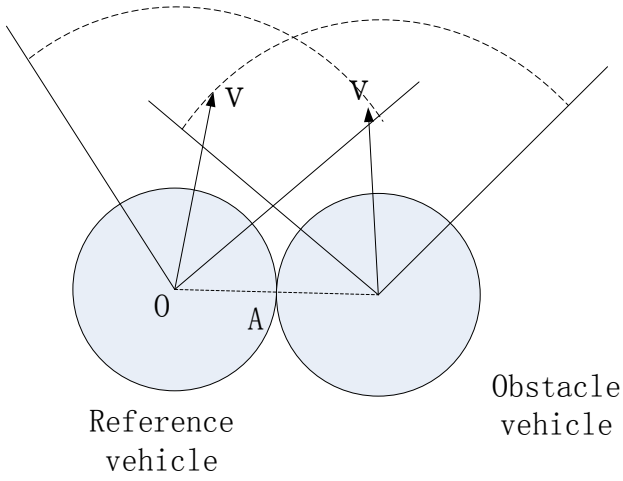


Figure 3.7 Example of the contacting point A when collision happens.

- The reference vehicle must have a relative velocity component $\overrightarrow{v_C}$ to the obstacle vehicle along the line of two centres (as shown in Figure 3.8), but they cannot detect each other; otherwise the collision will be avoided.

Figure 3.9(a) and Figure 3.9(b) illustrate two critical conditions when collision occurs. From Figure 3.9(a) to Figure 3.9(b), two vehicles are in each other's blind spot region (the region that the obstacle cannot be detected but collision may happen), the magnitude of $\overrightarrow{v_C}$ may range from $2v\cos(\alpha+\frac{\varphi}{2})$ to zero. Therefore, we have to find the probability of occurrence of the states of two vehicles within the range from Figure 3.9(a) to Figure 3.9(b). If the relative velocities are not in this range, the vehicles will never collide. We can see from the geometrical relationships that the included angle of two velocities is $180-(\varphi+2\alpha)$ in Figure 3.10(a), and 0 in Figure 3.10(b) respectively, so the probability that there is such a positive relative velocity within the range Figure 3.9(a) and Figure 3.9(b) is $\frac{180-(\varphi+2\alpha)}{360}$. This is one of the necessary conditions for collision.

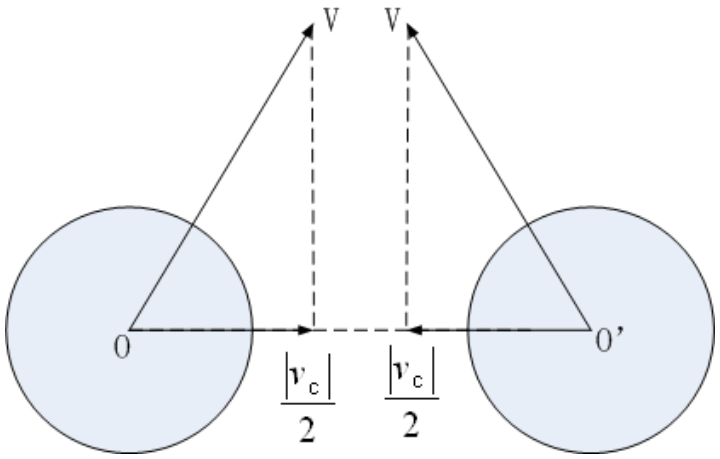
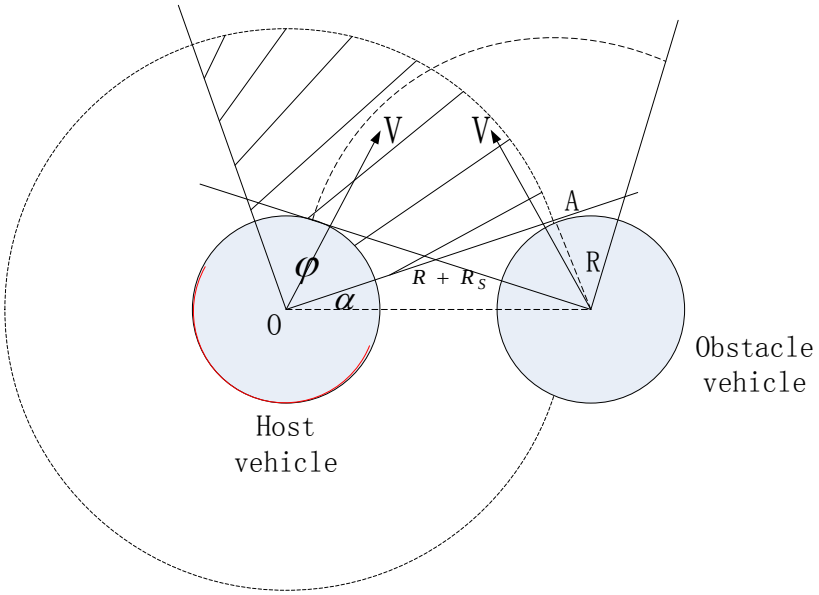
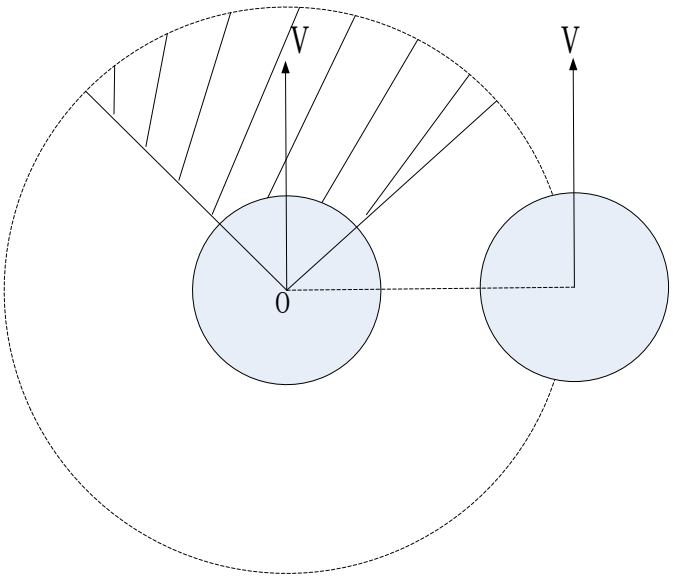


Figure 3.8 Relative velocity components along the line of two centres.

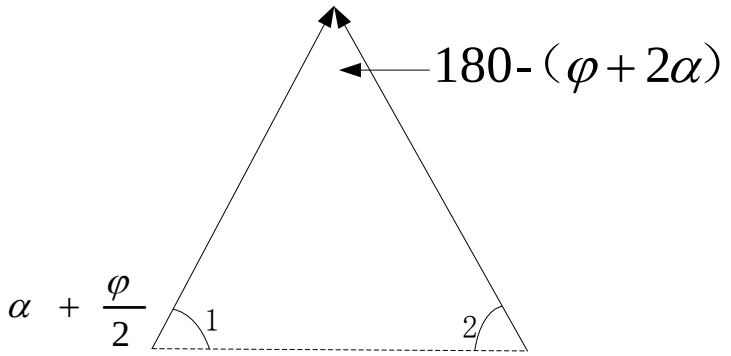


(a)

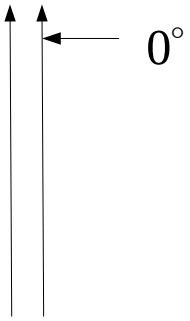


(b)

Figure 3.9 Two critical conditions of relative positions and postures when collision happens. (a) two vehicles just cannot detect each other (b) the velocities of two vehicles are parallel.



(a)



(b)

Figure 3.10 Included angles of velocities in two critical cases (a) included angle: $180-(\varphi+2\alpha)$ (b) included angle: 0

- Another necessary condition is that the contacting point must be within the blind spot. Figure 3.6 shows the critical condition that the host vehicle just cannot detect the obstacle one. Because the vehicles cannot move backwards, they will not collide with each other on the rear part $\overline{CDE}$. Therefore, the contacting point must be within angle θ, when the obstacle vehicle is not detected. Besides, because of the symmetry of vehicle, the angle range of blind spot is 2θ. Furthermore, the directions of the vehicles are decided randomly. Therefore, for the reference vehicle, the probability that the contacting point is within 2θ is $\frac{2\theta}{360}=\frac{180-\varphi-2\alpha}{360}$. The obstacle

vehicle can only collide with the reference vehicle from opposite sides, so the probability of collision for obstacle vehicle is $\frac{1}{2} \cdot \frac{180-\varphi-2\alpha}{360}$.

On the premise that two vehicles encounter, collision will occur only when both conditions shown above are satisfied, so the probability of collision is the product of the two probabilities:

$$P = \frac{180-(\varphi+2\alpha)}{360} \cdot \frac{180-\varphi-2\alpha}{360} \cdot \frac{1}{2} \cdot \frac{180-\varphi-2\alpha}{360}$$

$$= \frac{1}{2} \cdot \left(\frac{180-\varphi-2\alpha}{360}\right)^3 \quad (3.9)$$

where $\varphi + 2\alpha$ is the sensing angle.

3.3.2 Mathematical Formulation

After knowing the probability of collision *P* when two vehicles encounter, we also need to find the number of vehicles encountered by a reference vehicle in unit time. First, the number of vehicles that a reference vehicle encounters in unit time will be derived here, borrowing the idea from derivation of mean free path. Furthermore, the expected time of first collision can be obtained.

In section 3.1, we have introduced the concept of mean free path and its application in our study. It is only a general idea in physics in terms of microscopic particles. A model will be built according to the specific problem and macroscopic property of vehicles in this study. The derivation of mean free path will be used in this section to find the number of collisions between two vehicles.

Assume that there are a total of n vehicles with random initial positions

and orientations of velocities, within a rectangular area S. Similar to molecules, we consider one vehicle as the reference vehicle. Relative to this reference vehicle, all the other vehicles may be regarded as static obstacles. The density (number of vehicles per unit area) of the obstacles (excluding the reference vehicle) is

$$\rho = \frac{n-1}{S} \tag{3.10}$$

An effective vehicle radius $r(R, R_S)$ for computing the number of collisions is defined here. $r(R, R_S)$ is the equivalent radius of a vehicle that will result in the same probability of collision as that of the current model with radius R and R_S. The sensors reduce the probability of collision, so from the derivation of mean free path, it is obvious that the effective radius $r(R, R_S) < R$. Also, we can see that a collision occurs if the centres of two vehicles come within a distance $2r(R, R_S)$ of each other, which is shown in Figure 3.11. Similar to the derivation of mean free path, when a single vehicle zigzags in the confined area (see Figure 3.12), the virtual circle (see Figure 3.13) sweeps out a short rectangle of width $4r(R, R_S)$ between successive collisions. Because the vehicle model is in 2D, which is different from the molecule model, it can only sweep out a rectangle rather than a cylinder, as illustrated in section 3.1. A reference vehicle can move a distance $v\Delta t$ in time interval Δt, where v is the average speed of the reference vehicle. So the rectangle that the vehicle sweeps out is of area $4r(R, R_S)v\Delta t$. In unit time ($\Delta t = 1$), the rectangular area that is swept out is

$$A = 4 \cdot r(R, R_S) \cdot v \tag{3.11}$$

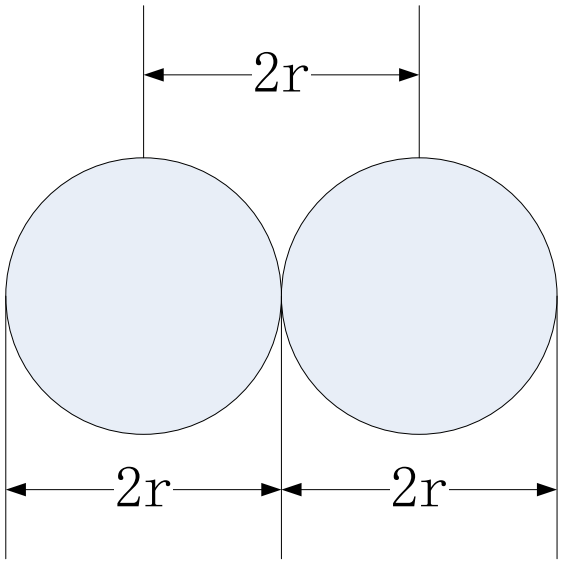


Figure 3.11 Distance between two centres of vehicles when collision happens.

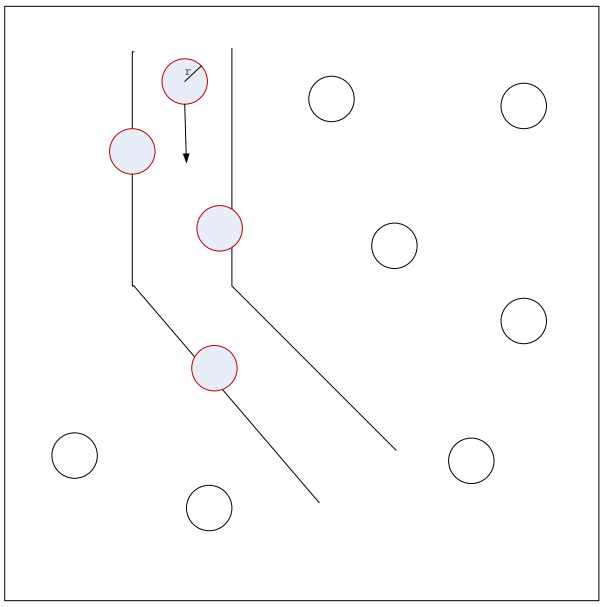


Figure 3.12 The area that is swept out by the reference vehicle when it moves in a confined area.

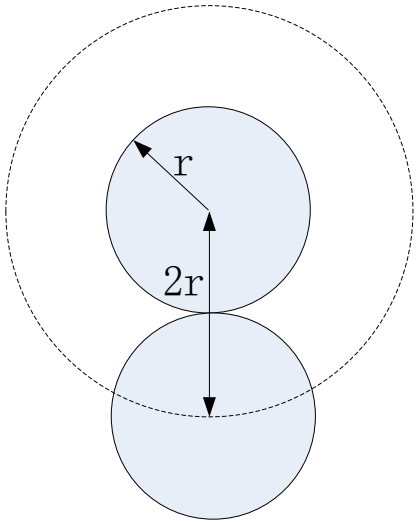


Figure 3.13 Critical condition of collision for vehicles.

So the number of vehicles that the reference vehicle will encounter in unit time is

$$m^* = \rho A = \frac{n-1}{S} 4r(R, R_S)v \tag{3.12}$$

Similar steps can be used to find $r(R, R_S)$. The radius of vehicle without sensor is called geometric radius. The area of the rectangle swept out by vehicle with geometric radius is $4 \cdot R \cdot v$. Therefore, the number of vehicles in this rectangle is $\frac{n-1}{S} 4Rv$. However, the sensing area will also sweep out some area (called avoided area, as shown in Figure 3.14), and the vehicles inside this area must be deducted, and it depends on the shape and size of sensing area. In Figure 3.14, If the centres of vehicles are within the area $(R + R_S)$, they will collide with the reference vehicle, and the maximum number of vehicles inside avoided area is $\frac{n-1}{S}(R + R_S)v$. So the number of vehicles encountered by the reference vehicle with equivalent radius is

$$\frac{n-1}{S} 4r(R, R_S)v = \frac{n-1}{S} 4Rv - \frac{n-1}{S}(R + R_S)v \tag{3.13}$$

So we get

$$r(R, R_S) = \frac{3}{4}(R - \frac{1}{3}R_S) \tag{3.14}$$

The expression for $r(R, R_S)$ will depend on different shape of vehicles and sensing area, so it must be decided according to the scenario being studied.

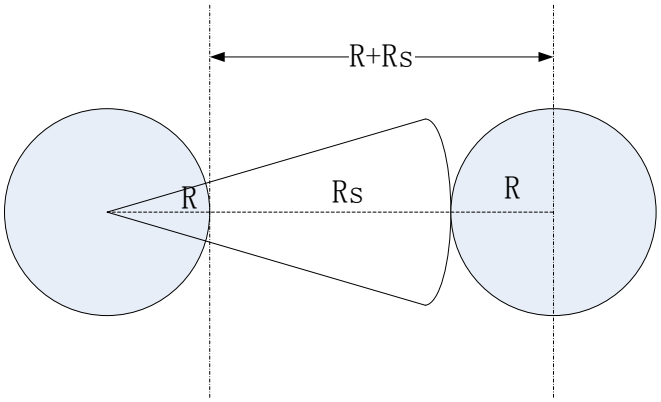


Figure 3.14 Collisions can be avoided if they are within the area $R + R_S$.

The assumption of the derivation is that only one vehicle is moving, and the others are static. If the static obstacle vehicles are also moving, the average speed v is the same as the average relative speed $\overline{v_{rel}}$, so the number of vehicles that the reference vehicle will encounter is

$$m_1 = \frac{n-1}{S} 4r(R, R_S)\overline{v_{rel}} = C_1 \frac{n-1}{S} 4r(R, R_S)v \tag{3.15}$$

where C_1 is a constant to convert average absolute velocity to average relative velocity, and $\overline{v_{rel}}$ is the average relative velocity between two vehicles.

The derivation above is based on the idea of mean free path of gas molecules, and in kinetic theory the volume of molecular gas molecule is negligible. However, for our problem the size of vehicles is not negligible compared with the size of the operation area. For molecular gas, as the size of

particle increases, the frequency of collision will also increase. Therefore, some modification is needed. The size of each vehicle is $\pi r^2(R, R_S)$, so we suppose that there is a term proportional to the ratio of total size of vehicles to the area of the workspace:

$$m_2 = C_S \frac{n\pi r^2(R,R_S)}{S} \tag{3.16}$$

where C_S is a constant. After modification, the number of vehicles that the reference one will encounter is

$$\begin{aligned} m &= m_1 \cdot m_2 \\ &= C_1 \frac{n-1}{S} 4r(R, R_S)v \cdot C_S \frac{n\pi r^2(R,R_S)}{S} \\ &= C_2 \frac{(n-1)n}{S^2} r^3(R, R_S)v \end{aligned} \tag{3.17}$$

where C_2 is a constant.

We have derived m, the number of vehicles that the host vehicle will encounter, but this is not the number of collision. In 3.3.1, we know that the probability of collision when two vehicles have been confirmed to encounter each other is P. Therefore, the number of collisions in unit time n_0 can be described as the product of m and probability P. So the number of collisions in unit time is

$$\begin{aligned} n_0 &= P \cdot m \\ &= \frac{1}{2} \cdot \left(\frac{180-\varphi-2\alpha}{360}\right)^3 \cdot C_2 \frac{(n-1)n}{S^2} r^3(R, R_S)v \\ &= C_3 \frac{(n-1)n}{S^2} r^3(R, R_S)v(180-\varphi-2\alpha)^3 \end{aligned} \tag{3.18}$$

where C_3 is a constant.

The time when first collision occurs is the reciprocal of the number of collisions per unit time, so the expected time of first collision T is a function of speed, FOV and number of vehicles:

$$T(n, v, \varphi) = \frac{1}{n_0} = C \frac{S^2}{r^3(R, R_S)} \frac{1}{(n-1)nv} \frac{1}{(180-\varphi-2\alpha)^3} \tag{3.19}$$

where C is a constant.

3.4 Analysis of Results

It can be seen from Eq.(3.19) that if $n = 1$, T will be infinite, which means that when there is only one vehicle in the area, it will never collide with other vehicles. Besides, if $v = 0$, T will also be infinite. When the vehicle has no speed, in other words, all of them do not move, collision will never happen. And if $\varphi + 2\alpha = 180°$, collision also cannot occur, because in this case the vehicles can always detect each other.

If we set v and φ to be fixed numbers, the time to first collision T is a function of the number of vehicles n,

$$T(n) = \frac{a}{(n-1)n} \tag{3.20}$$

where a is a constant decided by v, φ and constant C in Eq.(3.19).

When n is small, $T(n)$ is very large, but $T(n)$ decreases rapidly as n increases. For larger n s, $T(n)$ remains at a small decreasing rate. So, if we define a value T_{cr} as the critical time of collision, below which collision can be

deemed to occur instantaneously, the critical number of vehicles can be calculated.

Assume

$$M = C\frac{S^2}{r^3(R,R_S)} \tag{3.21}$$

in Eq.(3.19), and let $T \leq T_{\mathrm{cr}}$, we will get

$$\frac{1}{(n-1)n}\cdot\frac{M}{T_{cr}}\cdot\frac{1}{v}\cdot\frac{1}{(180-\varphi-2\alpha)^3} < 1 \tag{3.22}$$

Denote

$$\widetilde{M} = \frac{M}{T_{\mathrm{cr}}}\cdot\frac{1}{v}\cdot\frac{1}{(180-\varphi-2\alpha)^3} \tag{3.23}$$

where v and φ are fixed. The critical number of vehicles n is determined by solving Eq.(3.22)

$$n_{cr} = \frac{1}{2} + \frac{\sqrt{1+4\widetilde{M}}}{2} \tag{3.24}$$

which means if the number of vehicles is larger than n_{cr}, the expected time of first collision will be smaller than T_{cr}. In some related works, for example path planning, this can be a reference for maximum number of vehicles.

3.5 Simulation and Discussion

In this section, Monte Carlo simulation is carried out to verify the theory developed. The parameters will be specified, and the simulations are conducted in terms of three variables. Furthermore, the method to calculate the critical number of vehicles will be illustrated.

3.5.1 Simulation Environment

MATLAB is used as simulation tool in the study. The program is compiled and run on the software of MATLAB. According to the vehicle model and motion pattern of vehicles, the time to first collision in the confined area is obtained by Monte Carlo simulation. The simulation environment is introduced in Appendix III in detail.

3.5.2 Parameters of Vehicles and Workspace

Throughout the simulation study, the operation area is scaled to $x(\mathrm{m}) \in [-1,1], y(\mathrm{m}) \in [-1,1]$. Figure 3.15 shows the workspace which is attached to a reference frame. The workspace is a rectangular area, and its centre is located at the origin of coordinate frame. The positions of the vehicles in the workspace can be presented by the coordinates accordingly. The scale of the workspace can be changed according to the relative size of vehicles and workspace in cases. If the size of vehicles and the range of workspace are increased or decreased proportionally, the result in this study will keep the same. Furthermore, changing them disproportionally will result in different outcome, so some further study can be made.

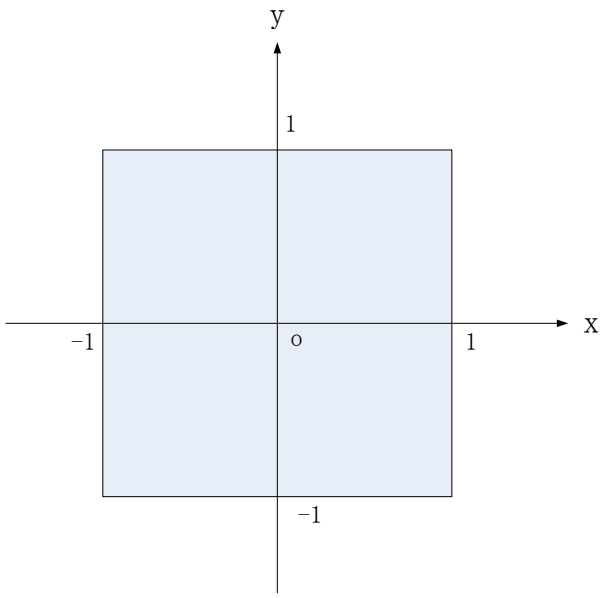


Figure 3.15 Diagram of workspace in coordinate system.

In the simulation experiments, the number of vehicles n varies from 5 to 46 (simulation step: 1). Speed v is from $0.2\mathrm{m/s}$ to $2\mathrm{m/s}$ (simulation step: $0.1\mathrm{m/s}$), and sensing angle φ is from $60°$ to $120°$ (simulation step: 2°), and the average time of first collision is recorded as the result. For each set of combination of n, v and φ, the simulation runs 1000 times to make the result more persuasive and accurate. Besides, we suppose that the parameters in simulations are $R_S = R$, $R = 0.025\mathrm{m}$, and then by calculation, α is determined to be 18.4°.

As the principle of computer calculation is discrete, the time step is set to $dt = 5 \times 10^{-3}\mathrm{s}$, and real time is the product of time step and number of steps. All the vehicles in the workspace will operate together in every time step, and after one step all the variables and commands will be updated.

When the distance between the centres of two vehicles is less than twice

of their radius, collision is considered to occur. Then the program will stop and the time cost from beginning can be calculated.

The process of collision can also be seen visually, and the interface is shown in Figure 3.16. Each circle represents a vehicle. We can see clearly that the program will stop when two vehicles collide.

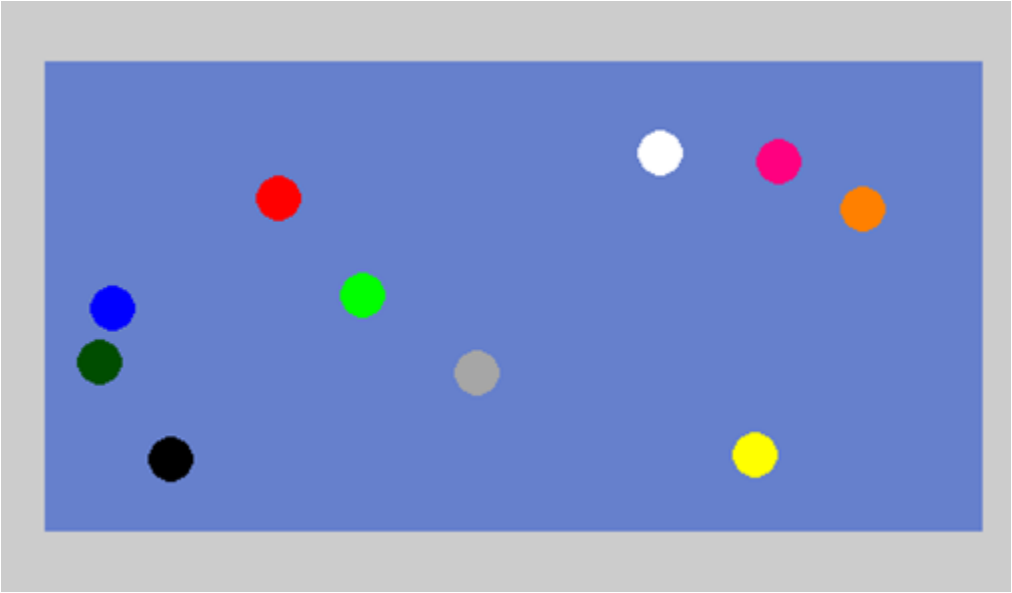


Figure 3.16 Visualized interface of the running program.

3.5.3 Flow Chart of Program

The flow chart of the program is shown in Figure 3.17. In every time step, collision between every two vehicles is checked. If collision occurs, the program will stop. This program will be repeated 1000 times for each set of condition.

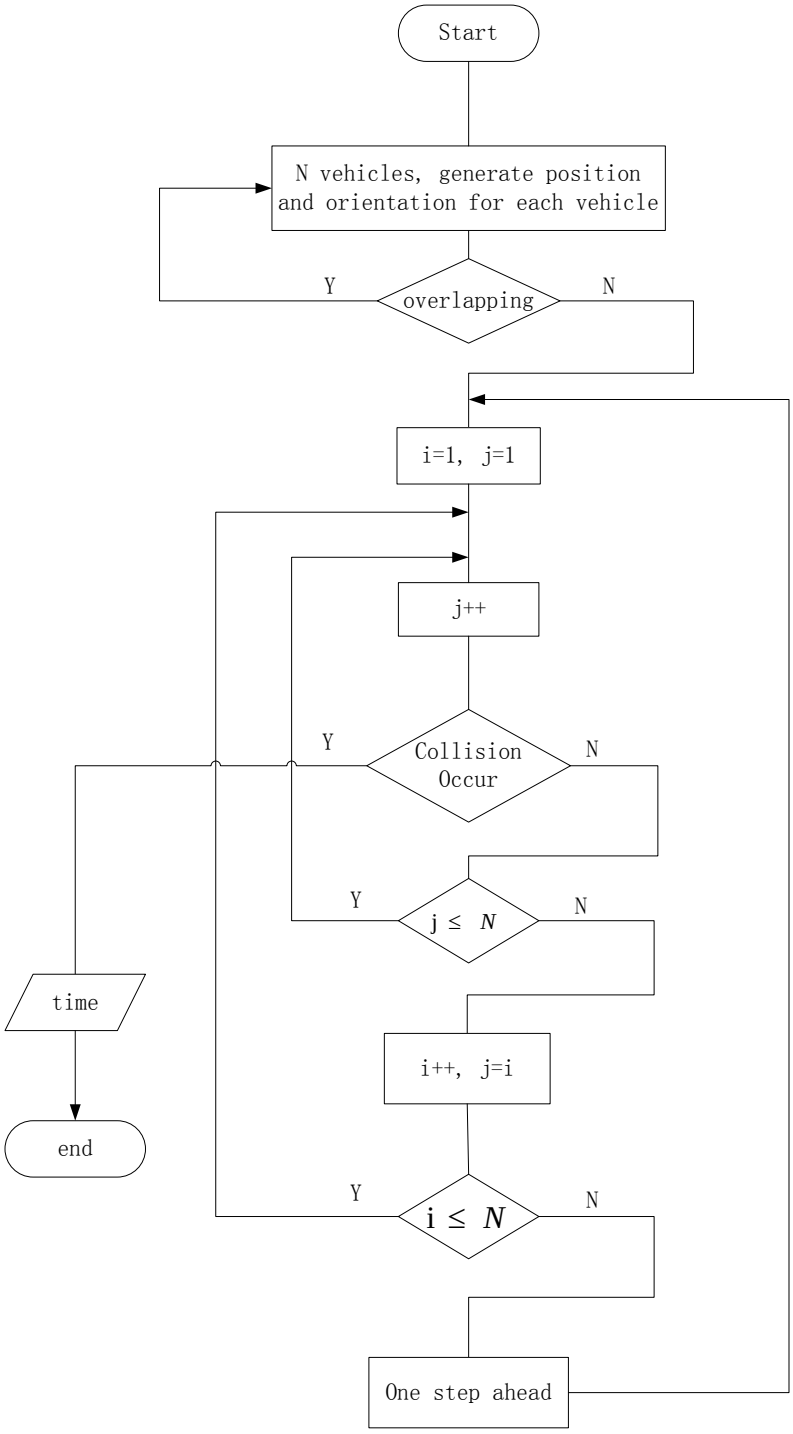


Figure 3.17 Flow graph of the program for calculating the time to first collision for vehicles with zero turn radius in a confined area.

3.5.4 Simulation Results

In this section, the simulation results will be revealed and be compared to the theory developed. In order to verify if the simulation result agrees with

Eq.(3.19), which is a function of n, v and φ in this study, the mean value of simulation results are fitted to the curves in terms of each variable, which will be shown in Eq.(3.25), Eq.(3.26) and Eq.(3.27). Besides, in order to see the goodness of fit between the simulation points and the theoretical curves, the coefficient of determination ${R_d}^2$ is provided [110], which is between 0 and 1. An ${R_d}^2$ value closer to 1 indicates that the curve has a better fit to the data.

Among lots of simulation conditions, we select three examples to see the results in three sorts of conditions. In each example, two variables are fixed to verify the form of the function in terms of the other variable.

3.5.4.1 Speed and Field of View Fixed

An example will be shown in this section in which velocity and FOV are fixed and only the number of vehicles varies. First, we select $v = 1\mathrm{m/s}, \varphi = 60°$, which are constants. According to Eq.(3.17), When v and φ are constants, the expected time is a function of n in expression

$$T(n) = \frac{a}{(n-1)n} \tag{3.25}$$

where a is a constant. The discrete points from simulation results can be fitted to the curve of Eq.(3.25). Figure 3.18 (a) shows the fitting curve and simulation points, where $a = 1060$. Coefficient of determination of the curve is ${R_d}^2 = 0.9955$, which is close to 1. The vertical line on each point represents the standard deviation of simulation data of each point. Figure 3.18 (b) shows the residuals between each data point and the corresponding point on the curve.

It can be seen from the upper figure that when the number of vehicles is relatively small, the time of first collision is relatively large, but it decreases rapidly at a decreasing rate as n increases. If n is large enough, the time becomes so short that it can be ignored, which means collision will happen instantaneously once operating.

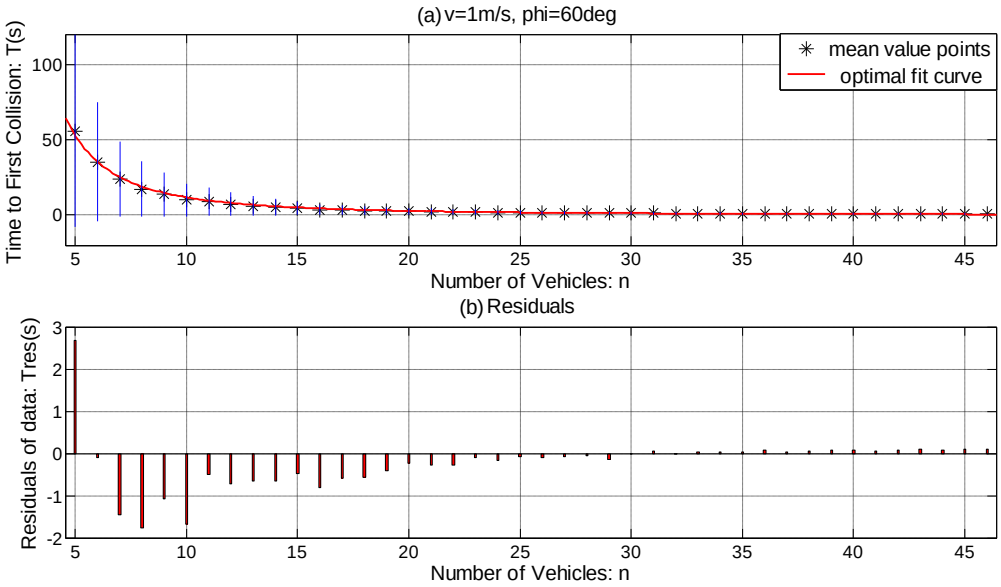


Figure 3.18 (a) Fit curve and residuals with respect to n when $v = 1\text{m/s}, \varphi = 60°$. (b) residuals of the fitting curve.

3.5.4.2 Speed and the Number of Vehicles Fixed

In this section, the simulation points are fitted to the function of the FOV of vehicles. $v = 1\text{m/s}, n = 14$ are selected as constants. Under the condition above, the expected time is a function of sensing angle φ with expression

$$T(\varphi) = \frac{b}{(143.2-\varphi)^3} \tag{3.26}$$

where b is a constant. The discrete points from simulation results can be fitted to the curve of Eq.(3.26). Figure 3.19 shows the fitting curve when $b = 3.498 \times 10^6$ and the residuals between each simulation point and the

corresponding point on the curve. Coefficient of determination of the curve is $R_d{}^2 = 0.9868$.

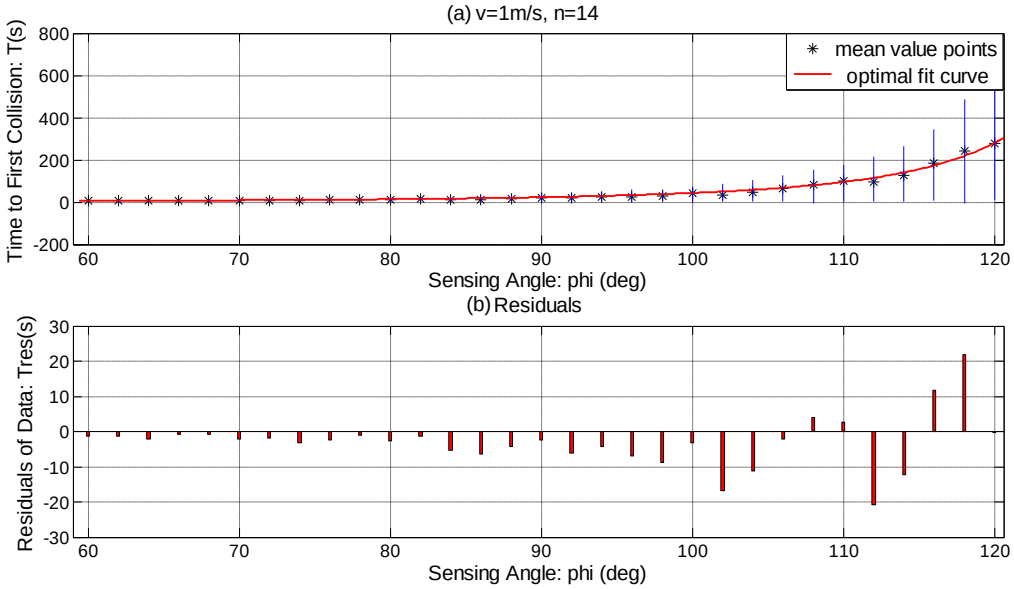


Figure 3.19 (a) Fit curve and residuals with respect to φ when $v = 1\text{m/s}, n = 14$. (b) residuals of the fitting curve.

3.5.4.3 FOV and the Number of Vehicles Fixed

In this section, the simulation points are fitted to the function of the speed of vehicles. $n = 20, \varphi = 90°$ are selected as constants. The expected time is a function of speed v with expression

$$T(v) = \frac{c}{v} \tag{3.27}$$

where c is a constant. The discrete points from simulation results can be fitted to the curve of Eq.(3.27). Figure 3.20 shows the fitting curve when $c = 9.183$ and the residuals between each simulation point and the corresponding point on the curve. Coefficient of determination of the curve is $R_d{}^2 = 0.9973$.

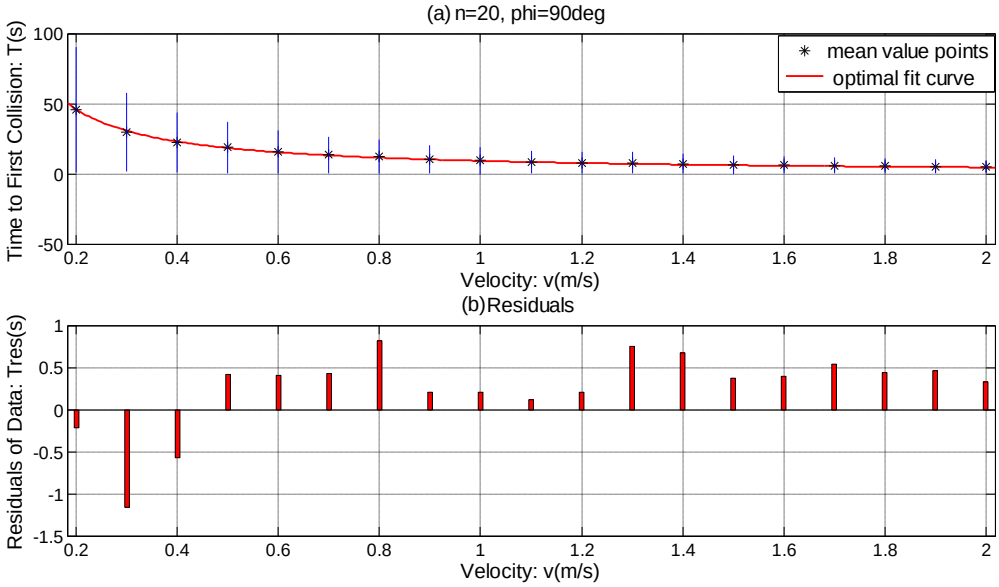


Figure 3.20 (a) Fit curve and residuals with respect to v when $n = 20, \varphi = 90°$. (b) residuals of the fitting curve.

3.5.5 Discussion

Among lots of simulation conditions, in section 3.5.4 we selected three examples to see the results in three sorts of conditions. The mean values and standard deviation of simulation data are shown in the figures and the mean values are fitted to the curves which are derived from the formula Eq.(3.19) developed in previous chapters.

In 3.5.4.1, the velocity and FOV of the vehicles are fixed, so the expected time to first collision is a function of n. From the fit curve, it can be seen that the time of first collision decreases with growing n at a decreasing rate. It means that when the number of vehicle increases in a confined area, the expected time of first collision will be shorter. Besides, while the number of vehicles n becomes larger and larger, the effect of n to the time will be less. If the number of vehicles becomes very large, the average distance between them

will be short, and collision almost occur instantaneously. Figure 3.18 shows that Eq.(3.25) can fit the data very well. Coefficient of determination of the curve is ${R_d}^2 = 0.9955$, which is very close to 1. Most of the residuals are around 5%. Besides, the standard deviation decreases when the time of first collision becomes smaller, because the average distance between vehicles are smaller, and the maximum time of first collision must be shorter.

In 3.5.4.2, the time to first collision is a function of FOV. From the simulation result we can see that the expected time to first collision will increase at an increasing rate when FOV becomes larger. Coefficient of determination of the fit curve is ${R_d}^2 = 0.9868$, so the simulation data has a good fit. The residuals are also around 5%.

In 3.5.4.3, the time to first collision is a function of speed v. The time of first collision decreases with speed v at a decreasing rate. Coefficient of determination of the curve, ${R_d}^2 = 0.9973$, and the residuals in the figure also show the good fit between data and fit curve.

Consequently, the expected time to first collision will decrease when the number of vehicles and the velocity of vehicles increase, but the time will increase when FOV becomes larger, which is matching with our intuition. In Figure 3.18, Figure 3.19 and Figure 3.20, the standard deviations are relatively large, because that the positions and velocities of the vehicles are generated by the program randomly. For example, when the distance between two vehicles is very small and the relative velocities tend to result in collision, collision may happen in a short time regardless of other factors. Under this circumstance, the expected value can reveal the trend properly, which is used

in fitting. From all the figures, it can be seen that the curves can fit the simulation results very well and the coefficient of determination of the curves are very close to 1. Most of the residuals maintain at around 5%. Therefore, the theory developed above is verified.

After we obtain the constants of fit curves, the value of M in Eq.(3.22) can be determined, which is calculated from the parameters of fit curves (a, b and c). The value of M calculated from the three situations are as below:

$$M = 6.10 \times 10^8 \text{m} \tag{3.28}$$

$$M = 6.37 \times 10^8 \text{m} \tag{3.29}$$

$$M = 5.25 \times 10^8 \text{m} \tag{3.30}$$

Eq.(3.28), Eq.(3.29) and Eq.(3.30) are calculated from section 3.5.4.1, 3.5.4.2 and 3.5.4.3 respectively. We can see that they are of the same order of magnitude, and only small differences exist because of the inevitable error of fitting. Furthermore, the critical number of vehicles n_{cr} will be found when the critical time T_{cr} is defined. For example, if we define $T_{\text{cr}} = 5\text{s}$, we can get $\widetilde{M}$ after obtaining the value of M. Some critical numbers of vehicles n_{cr} for different v (m/s) and φ (degree) from Eq.(3.24) are shown in Table 3.1 and Figure 3.21. The results will be useful in planning the number of vehicles in a confined area.

Table 3.1 Critical number of vehicles when $T_{\text{cr}} = 5\text{s}$

n_{cr} φ / v	60	65	70	75	80	85	90	95	100	105	110	115	120

0.1	47	51	56	63	70	79	90	105	123	148	182	232	311
0.2	33	36	40	44	50	56	64	74	87	105	129	165	220
0.3	27	30	33	37	41	46	53	61	71	86	106	135	180
0.4	24	26	29	32	35	40	46	53	62	74	92	117	156
0.5	21	23	26	29	32	36	41	47	56	67	82	104	140
0.6	20	21	24	26	29	33	37	43	51	61	75	95	128
0.7	18	20	22	24	27	30	35	40	47	56	69	88	118
0.8	17	19	21	23	25	29	33	38	44	53	65	83	111
0.9	16	18	19	22	24	27	31	36	42	50	61	78	104
1.0	15	17	18	20	23	26	29	34	40	47	58	74	99
1.1	15	16	18	20	22	25	28	32	38	45	56	71	94
1.2	14	15	17	19	21	24	27	31	36	43	53	68	90
1.3	14	15	16	18	20	23	26	30	35	42	51	65	87
1.4	13	14	16	17	19	22	25	29	34	40	49	63	84
1.5	13	14	15	17	19	21	24	28	33	39	48	61	81
1.6	12	14	15	16	18	21	23	27	32	38	46	59	79

1.7	12	13	14	16	18	20	23	26	31	37	45	57	76
1.8	12	13	14	16	17	19	22	25	30	36	44	56	74
1.9	11	13	14	15	17	19	21	25	29	35	43	54	72
2.0	11	12	13	15	16	18	21	24	28	34	41	53	70

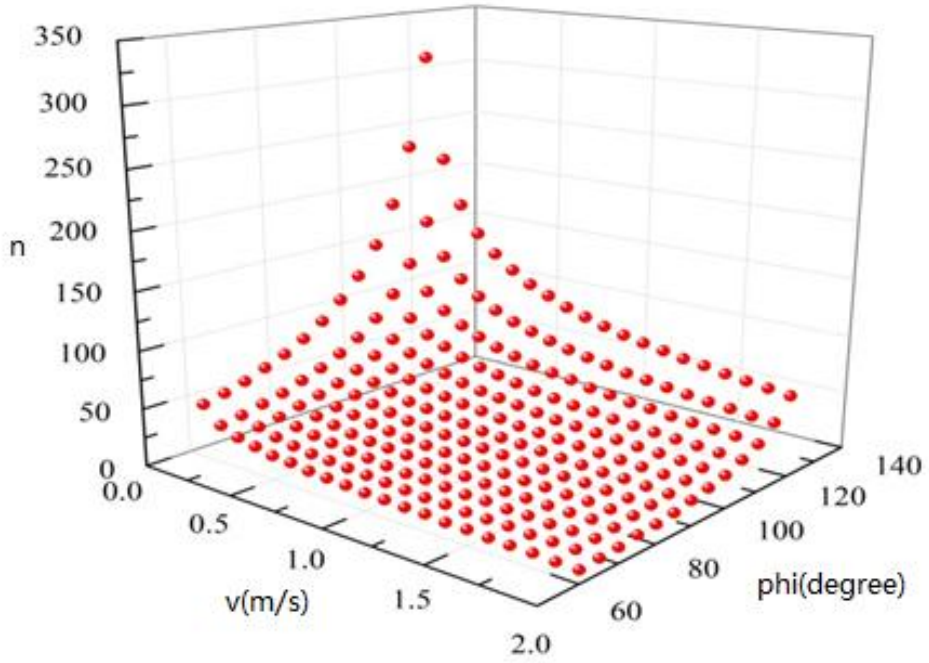


Figure 3.21 Critical number of vehicles when $T_{\mathrm{cr}} = 5\mathrm{s}$ with respect to different speeds and FOVs.

3.6 Conclusion

In this chapter, we studied the time to first collision of UxVs with zero turn radius in a confined area. Safety is a basic premise of the operations of vehicles, so the study in this chapter aims at finding the factors affecting the

probability of collision. In order to check the influence of these factors, we commence with a simple case, vehicles with zero turn radius in a confined area. In this study, based on the simple model of vehicle proposed, the formula to calculate the expected time to first collision of multi-vehicle system in a confined area is derived. The expected time to first collision will decrease at a decreasing rate while the number of vehicles and the velocity of vehicles increase, but it will increase at an increasing rate when FOV increases.

Monte Carlo simulations were conducted, by setting two of three variables n, v and φ as constants, to get the expected time to first collision and standard deviation under each different condition. Consequently, the simulation data are fitted to the curve derived mathematically, and the goodness of fit was checked. The simulation results validate the theory developed, and all the results agree with our intuition. The model of the vehicle can be changed for different situations and the variables affecting collision can also be different. The results will be useful as a reference for UxVs operations in confined areas. In the next chapter, we will study another vehicle model with a different collision avoidance method, velocity obstacle.

Chapter 4. Time to First Collision for Dubins' Vehicles with Non-zero Turn Radius in a Confined Area

In Chapter 3, our study was on a simple vehicle model. In this chapter, we will study the time to first collision in a confined area for Dubins' vehicles with collision avoidance based on the Velocity Obstacle method. The Velocity Obstacle method is found in many applications in avoiding obstacles for robotic vehicles. We will derive the formula of the time to first collision in a confined area, and the relation between different variables. System parameters can then be selected to fulfill task requirements before actual operation. In this chapter, the expected time of the first collision within the workspace will be expressed as a function of the kinematic variables of Dubins' vehicles, which is derived based on the concept of Mean Free Path. Furthermore, the formula will be verified and the parameters can be approximated by Monte Carlo simulations. Critical number of vehicles is also derived.

4.1 Introduction to Velocity Obstacle

For the operation of multiple vehicles, collision avoidance is a basic safety requirement. One of the widely used methods to detect and avoid moving obstacles is the Velocity Obstacle approach, which will be applied in this study. A reference vehicle can easily detect a potential collision once it knows the position and velocity of an obstacle, assuming that the obstacle vehicle will

continue moving in its current velocity. The main idea of Velocity Obstacle will be illustrated in the following.

The vehicles are supposed to be disk-shaped, and move in a 2D space. Assume that A and B are two vehicles, and the positions of the centres of them are P_A and P_B. Let their velocities be v_A and v_B respectively. The relative velocity of B to A is v_{AB}, and the relation is

$$v_{AB} = v_B - v_A \tag{4.1}$$

If the radii of the disks are r_A and r_B respectively, we define an equivalent radius r_{AB}, which is the sum of the two radii

$$r_{AB} = r_A + r_B \tag{4.2}$$

The model of two vehicles with radii r_A and r_B can be equivalent to that of one vehicle with radius r_{AB} and the other one is a mass point, when we study the collision conditions and avoidance in this study. The new disk with radius r_{AB} is called AB. Let $\lambda(P, v)$ denote the ray that start from P along the direction of v, so [52]

$$\lambda(P, v) = \{P + vt | t > 0\} \tag{4.3}$$

From the centre of a vehicle P_B we can draw two lines that are tangent to the equivalent disk with radius r_{AB} at point P_A, and a velocity obstacle VO_A^B will be formed, as shown by the shaded area in Figure 4.1. Velocity Obstacle is defined as

$$VO_A^B = \{v_{AB} | \lambda(P_A, v_{AB}) \cap AB \neq \emptyset\} \tag{4.4}$$

If the extension line of v_{AB} is within the velocity obstacle VO_A^B, B will collide

with A at some point in some future time. If not, they will not collide. The geometrical illustration is shown in Figure 4.1, which shows the case where collision will happen. Usually, we want to change the velocity of B to avoid collision and keep the velocity of A unchanged, so the Velocity Obstacle of B is found, so that we can determine whether collision will happen through velocity of B, and find a proper velocity of B to avoid collision that is happening. The velocity obstacle of B in Figure 4.2 is $VO_A^B(v_B)$. Because the velocity of B is

$$v_B = v_{AB} + v_A \tag{4.5}$$

$VO_A^B(v_B)$ can be demonstrated using formula

$$VO_A^B(v_B) = VO_A^B \oplus v_A \tag{4.6}$$

where $\oplus$ is Minkowski sum.

If the velocity of vehicle B at current time is within the velocity obstacle of B, in other words when $v_B \in VO_A^B(v_B)$, the vehicles will collide with each other, assuming that they move with current velocities. Therefore, the vehicle B needs to select a new velocity outside the velocity obstacle of B, i.e. $v_B \notin VO_A^B(v_B)$, so that collision will not happen.

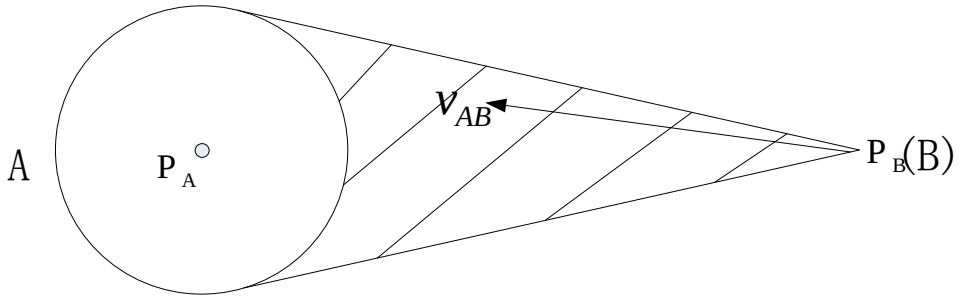


Figure 4.1 The diagram of velocity obstacle

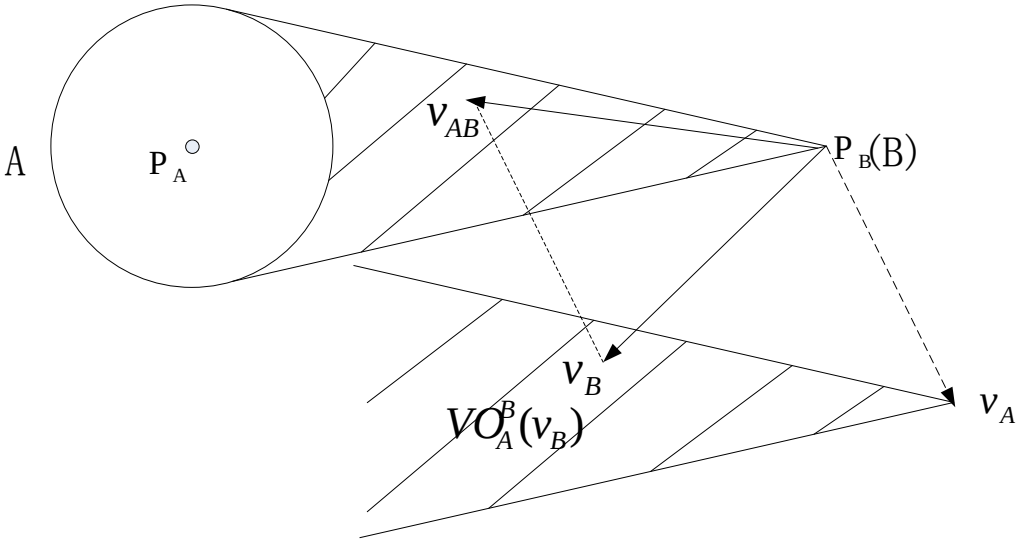


Figure 4.2 Velocity Obstacle of v_B

4.2 Model of Dubins' Vehicle

The model of Dubins' vehicle will be applied in this study. Dubins' vehicle is a kind of nonholonomic vehicle, which was proposed by Lester Eli Dubins in 1957 [101]. The vehicle is assumed to be a rigid body moving in a plane, and can only move forward with constant speed, and has a maximum acceleration,

which is always perpendicular to the direction of motion. The vehicle therefore has a minimum turn radius [101], and it is assumed that the vehicle only turns with the minimum turning radius. In the Cartesian coordinate system $(x, y, \theta)^T$, let $(x, y)^T$ denote the position of the centre of the vehicle, and θ denote the orientation of velocity with respect to x axis. The kinematic equations of Dubins' vehicle is as follow [111]

$$\begin{cases} \dot{x} = v\cos\theta \\ \dot{y} = v\sin\theta \\ \dot{\theta} = \omega \end{cases} \tag{4.7}$$

where v is the linear speed, and ω is the angular speed. In addition, the angular speed can be represented by

$$\omega = a/v \tag{4.8}$$

where a is the maximum acceleration of the vehicle. We can see that a and v are the only control variables for a vehicle.

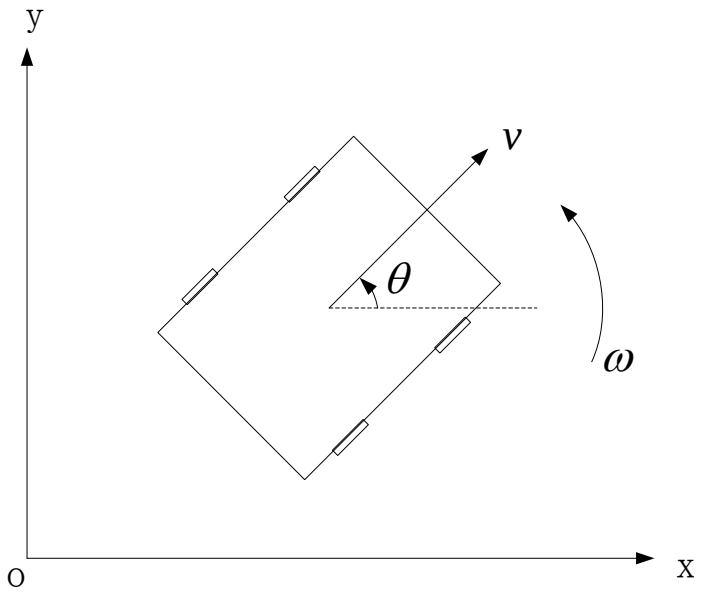


Figure 4.3 Dubins' vehicle in a coordinate system.

4.3 Motion Pattern of Vehicles

In a common workspace, unmanned vehicles (UxVs) require sufficient time to detect and resolve an imminent collision, as mentioned in last chapter. Therefore, a vehicle is always given a safety distance, and the distance between the vehicle and any obstacle must be greater than the safety distance. The proposed model of a vehicle in this study is a disk with radius R_{ob}, which is the safety distance of the vehicle. When we study the relative motion of two vehicles with safety distances R_a and R_b respectively, the model can be equivalent to that one vehicle, which is called reference vehicle, has safety distance $R_a + R_b$, and the other one is regarded as a mass point. The equivalent safety distance $R_a + R_b$ is denoted as R_{ab}. As all the vehicles share the same model, the equivalent safety distance R_{ab} is twice the radius of vehicle R_{ob}. The equivalent safety distance will be used in the following analysis.

Now, the proposed motion of vehicles will be introduced below. The motion of a Dubins' vehicle can be separated into two kinds, and the motion is illustrated in Figure 4.4. The first kind of motion is rectilinear motion, and the second kind is turning motion when an obstacle is detected. The vehicles move in constant speeds throughout the motions. In the second kind of motion, the vehicle needs to turn a certain angle at the maximum acceleration to avoid an obstacle vehicle. Once two vehicles detect each other, they will turn simultaneously to a certain angle. Based on the kinematic properties of the vehicles above, the expected time to the first collision in a confined area will

be derived. The two kinds of motions are described in the following.

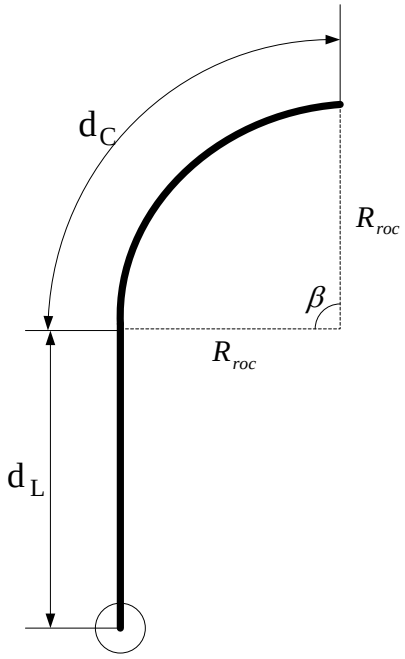


Figure 4.4 Two kinds of motions for the vehicles in this study: rectilinear motion and turning motion.

4.3.1 Rectilinear Motion

4.3.1.1 Rectilinear Motion without Considering Collision Avoidance

The basic motion of the vehicles is rectilinear motion. While working in the workspace, a vehicle will move in a straight line until it finds an obstacle within its safety distance. The expected distance of rectilinear motion is noted as d_L, and d_L can be derived referring to the derivation of Mean Free Path (MFP). In three-dimensional space, the vehicles will sweep out a cylinder when they move in a straight line, which was explained in detail in previous chapters. The same as the studies in last chapter, the vehicles also move in

only two dimensions in this study, so the detecting area of the reference vehicle can only sweep out a rectangle, and the vehicles inside the rectangle will collide with it, as shown in Figure 4.5. Because the vehicle will turn instantaneously when it finds an obstacle within its detecting area, the average distance that the vehicle travels to find the first obstacle will be the length of rectilinear motion d_L. Therefore, the derivation of d_L will be similar to the length of path that is derived in free motion. So first we will derive the length of rectilinear motion d_L. We assume that the speed of the reference vehicle is v, and an obstacle will be detected if the distance between the reference vehicle and the obstacle is less than $R_{ab} + 2R_{roc}$, where R_{roc} is the radius of curvature that will be introduced in the next section, and the value of R_{roc} is

$$R_{roc} = v^2/a \tag{4.9}$$

Therefore, the area of the rectangular detecting area that is swept out by the reference vehicle within time Δt is

$$A = 2(R_{ab} + 2R_{roc}) \cdot v\Delta t \tag{4.10}$$

In order to calculate the number of vehicles in this area, we let the density of obstacle vehicles in a workspace S be ρ, so the number of obstacle vehicles within the rectangular area is

$$N_r = A \cdot \rho \tag{4.11}$$

If the reference vehicle moves along straight line, the distance is $v\Delta t$ in time interval Δt. Along this line, the number of vehicles that are encountered is N_r, so the actual distance of rectilinear motion is $v\Delta t/N_r$. Therefore, the average distance of rectilinear motion d_L is

$$d_L = \frac{v\Delta t}{N_r} = \frac{1}{2(R_{ab} + 2R_{roc})\rho} \tag{4.12}$$

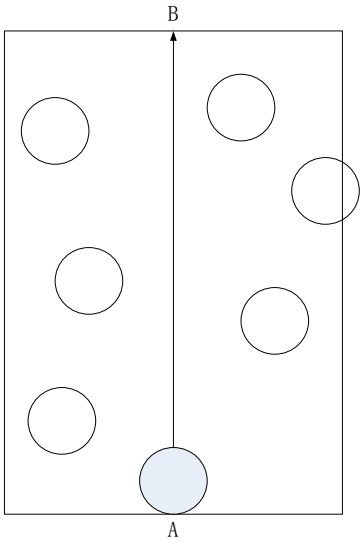


Figure 4.5 The rectangle swept out by a vehicle in 2D.

4.3.1.2 Average Distance of Rectilinear Motion

From the analysis above, we know that a vehicle will begin to avoid an obstacle vehicle once the obstacle vehicle is within the detecting area. In fact, most of these obstacle vehicles will not collide with the reference vehicle, because the relative velocity between them may not tend to draw them closer. We need to exclude the case where the vehicles do not collide. Velocity obstacle method is used to determine whether collision will happen. Now, the calculation will be introduced. In Figure 4.6, A is the reference vehicle, and B is the obstacle vehicle. We have introduced the basic principle of Velocity Obstacle, so here we only show the calculation of the specific problem. If the velocity of B relative to A is in the area of the cone, which is subtended by the disk with radius of equivalent safety distance and two tangent lines, this two vehicles that are assumed to continue moving with current velocities will

collide. The probability that the direction of relative velocity is within the cone need to be found, supposing that all the directions of the vehicles are random. In order to calculate the probability P_{cone} that the direction of relative velocity is within the cone, let the distance between A and B be d_{AB}, and half angle of the cone be α_{cone}, as shown in Figure 4.6. According to the geometrical and mathematical relations between the variables, the relation between α_{cone} and d_{AB} is

$$\sin \alpha_{cone} = R_{ab}/d_{AB} \tag{4.13}$$

so the angle α_{cone} is

$$\alpha_{cone} = sin^{-1}(R_{ab}/d_{AB}) \tag{4.14}$$

The probability P_{cone} that the relative velocity is within the cone is the ratio of the angle with the cone and the total angle, which is

$$P_{cone} = 2\alpha_{cone}/2\pi \tag{4.15}$$

Substitute Eq.(4.13) into Eq.(4.14), we can get

$$P_{cone} = \sin^{-1}(R_{ab}/d_{AB})/\pi \tag{4.16}$$

The relative distance d_{AB} between A and B depends on the position of the obstacle vehicle, in other words d is a variable, so the probability P_{cone} will also change with different d_{AB}. One effective way is to find the mean probability P_{mean}. The maximum distance between A and B is the detecting radius $R_{ab} + 2R_{roc}$, beyond which the obstacle cannot be detected. The least distance is the critical safety distance R_{ab}, below which the vehicles have already collided.

In mathematics, suppose that $y(x)$ is a continuous function of x on a closed interval $[a, b]$, the average value of function $y(x)$ from $x = a$ to $x = b$ is the integral of $y(x)$ over $[a, b]$ divided by the length of the interval, which is

$$\bar{y} = \frac{1}{b-a}\int_a^b y(x)dx \tag{4.17}$$

Therefore, based on the definition of the mean value of function, the mean probability in this study is:

$$P_{mean} = \frac{\int_{R_{ab}}^{R_{ab}+2R_{roc}} Pdx}{2R_{roc}} = \frac{\int_{R_{ab}}^{R_{ab}+2R_{roc}} \sin^{-1}(R_{ab}/x)dx}{2\pi R_{roc}} \tag{4.18}$$

So, for each vehicle in the detecting area, the probability of collision is P_{mean}, and only some of them will collide with the reference vehicle. The average number of obstacle vehicles should be smaller than N_r. The number of obstacle vehicles in the rectangular area N_r in Figure 4.5 should be multiplied by the probability of collision P_{mean}, so that we can get the number of obstacle vehicles that may collide with the reference vehicle. Consequently, the distance of rectilinear motion d_L also needs to be modified as:

$$d_L = \frac{v\Delta t}{N_r P_{mean}} = \frac{1}{2(R_{ab}+2R_{roc})\rho P_{mean}} = \frac{\pi R_{roc}}{(R_{ab}+2R_{roc})\rho \int_{R_{ab}}^{R_{ab}+2R_{roc}} \sin^{-1}(R_{ab}/x)dx} \tag{4.19}$$

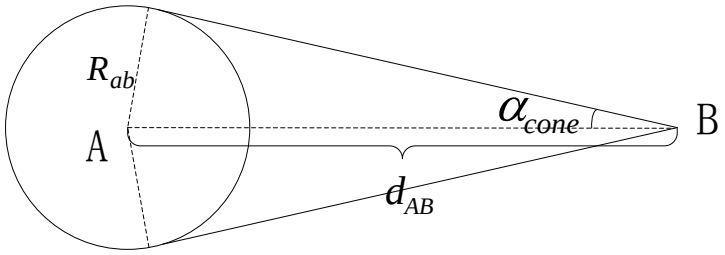


Figure 4.6 Velocity obstacle between vehicles A and B.

4.3.2 Turning motion

The other kind of motion is turning motion, which is the arc with length of d_C in Figure 4.4. If the reference vehicle detects that an obstacle needs to be avoided, when it moves along a straight line, we let the vehicle turn a certain angle β with the maximum acceleration. Therefore, from the geometrical relation, we can easily get the length of the curve

$$d_C = 2\pi R_{roc} \frac{\beta}{2\pi} = \beta R_{roc} \tag{4.20}$$

4.4 Derivation of Formula for Mean Time to the First Collision

4.4.1 General Formulation

In this section, we will derive the formula of the time to first collision in a confined area, based on the model we developed above. The motion of vehicles can be concluded as below: a vehicle moves in straight line until it finds an obstacle vehicle that needs to be avoided. The reference vehicle will

turn a certain angle to avoid collision. After a vehicle finishes turning, it continues to move in straight line, and repeats the motion until it collides with some obstacle that cannot be avoided. The procedure from beginning a rectilinear motion to completing a turning motion is defined as a cycle, so the length of a cycle is the sum of the expected distances of two motions:

$$d_{cycle} = d_L + d_C \tag{4.21}$$

The speed of the vehicles is constant. The distance that a vehicle moves in unit time is v, so the number of cycles in unit time is:

$$N_{cycle} = \frac{v}{d_{cycle}} \tag{4.22}$$

Therefore, if the expected number of collisions in each cycle is known, the expected number of collisions in unit time can be derived, so that we can know the time to first collision. In our model, collision can only happen during turning in each cycle, because the reference vehicle will turn instantaneously, if it is found that an obstacle needs to be avoided while the vehicle is moving straight forward.

Referring to the derivation of mean free path again, we can get the number of collisions while a vehicle is turning. A vehicle with radius R_{ab} can sweep out a region while it is turning, which is marked as shaded area in Figure 4.7, and it will collide with the obstacle vehicles that are inside the region. The size of the shaded area is:

$$S_C = \frac{\beta}{2\pi}[\pi(R_{roc} + R_{ab})^2 - \pi(R_{roc} - R_{ab})^2] = 2\beta R_{roc} R_{ab} \tag{4.23}$$

So the number of collisions in a cycle is:

$$N_{collision} = S_C \cdot \rho \tag{4.24}$$

Up to now, we can calculate the number of collisions in unit time N, which is multiplication of the number of cycles in unit time and the number of collisions in each cycle

$$N = N_{collision} \cdot N_{cycle}$$

$$= \frac{v}{d_{cycle}} S_C \cdot \rho \tag{4.25}$$

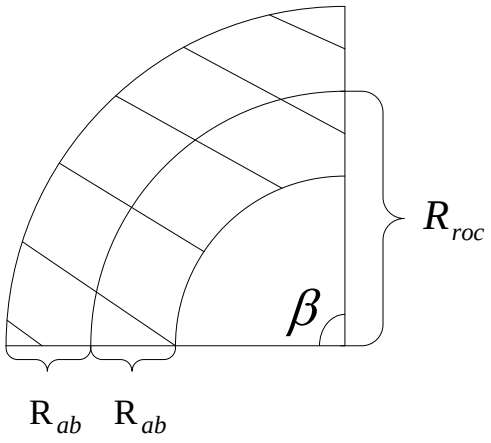


Figure 4.7 Diagram of turning motion. The shaded area is swept out by the reference vehicle.

4.4.2 Variables and Parameters

In the derivation above, we assume that only the reference vehicle is moving, so the average speed v should be replaced by the relative speed v_r when all the vehicles move. When the velocities of all the vehicles are random, the relation between two vehicles can be described by Figure 4.8. v_1 and v_2 are any two random velocities, and the average relative velocity is the difference between them

$$\bar{v}_r = v_2 - v_1 \tag{4.26}$$

The magnitude of the average relative velocity is

$$|\bar{v}_r| = \sqrt{\bar{v}_r \cdot \bar{v}_r}$$

$$= \sqrt{(v_2 - v_1) \cdot (v_2 - v_1)}$$

$$= \sqrt{|v_2|^2 + |v_1|^2 - 2v_1 \cdot v_2} \quad (4.27)$$

Because v_1 and v_2 are random and independent, the inner product of them is zero ($v_1 \cdot v_2 = 0$). The average magnitude of relative velocity is

$$|\bar{v}_r| = \sqrt{|v_2|^2 + |v_1|^2} \quad (4.28)$$

We can use the average velocity v to substitute v_1 and v_2, since they are randomly selected.

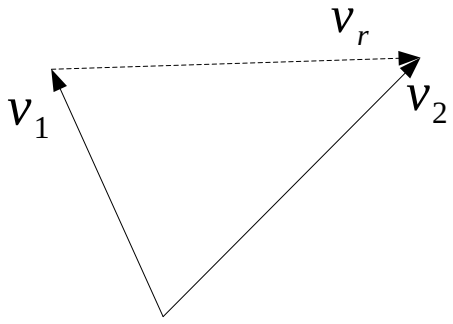


Figure 4.8 Relationship between the relative velocity and two random velocities.

Therefore, the relation between v and v_r is:

$$v_r = \sqrt{2}v \quad (4.29)$$

In addition, the density of vehicles ρ also appears in formula Eq.(4.24), so we need to find the expression of ρ. In theory, the density should be

$(n-1)/S$, where n is the number of vehicles in the workspace, and S is the area of the workspace, if the area of the workspace approaches infinite. However, the effect of the boundaries of the workspace is non-negligible in this study, so the expression of density needs to be revised. We suppose that the density of the vehicles is

$$\rho = C^{-1}(S, R_{ob}) \cdot \frac{n-1}{S} \tag{4.30}$$

where $C(S, R_{ob})$ is a function of the area of the workspace and the size of vehicles that we suppose to make up the effect of boundaries.

$C(S, R_o)$ is a term that reflects the effect of the boundaries of the workspace, so we can deduce the form of $C(S, R_{ob})$ based on its properties. Obviously, when the area of the workspace S and the number of vehicles n are infinite, $C(S, R_{ob})$ will approach 1. In other words, if the area of the workspace is huge enough compared with the size of the vehicles, $C(S, R_{ob})$ is 1. In our assumption, a possible form of $C(S, R_{ob})$ is exponential function, so we suppose that

$$C(S, R_{ob}) = \alpha^{-\beta(\frac{R_{ob}}{S})^k} \tag{4.31}$$

where α, β and k are constants, and $\alpha \in (1, \infty), k > 0, \beta > 0$. In this function, when the area of the workspace S approaches ∞, the value of $C(S, R_{ob})$ is 1, so the area of the workspace will not affect the expected time of the first collision T. When $S = 0$, the value of $C(S, R_{ob})$ approaches 0, and in Eq.(4.29) the density approaches infinite. Therefore, the time to first collision T, which is the reciprocal of the number of collisions in unit time, will also approaches 0.

Now, the formula of the number of collisions in unit time N can be derived, so the expected time of the first collision T in the workspace is:

$$T = 1/N \tag{4.32}$$

In order to simplify the calculation, we let the turning angle β be $\frac{\pi}{2}$ in this study. Consequently, based on the analysis above, Eq.(4.9) to Eq.(4.32), so the expression of T is as follow:

$$T = C_1 \cdot \frac{S^2 a}{\sqrt{2} v (4v^2 + R_{ab}\, a) R_{ab} \int_{R_{ab}}^{R_{ab}+2R_{roc}} \sin^{-1}\left(\frac{R_{ab}}{x}\right) dx} \cdot \frac{1}{(n-1)^2} + C_2 \cdot \frac{S}{2\sqrt{2} R_{ab}\, v} \cdot \frac{1}{n-1} \tag{4.33}$$

where $C_1 = C^2(S, R_{ob})$, $C_2 = C(S, R_{ob})$. Now, we have got the formula of the time to first collision in a confined area. This formula will be analyzed in the next section.

4.5 Analysis of Results

4.5.1 Approximation of Integration

Formula Eq.(4.33) is a function of average speed v, acceleration a, the number of vehicles n, the area of workspace S, the size of vehicle R_{ob}, and safety distance R_{ab}. The time to first collision can be calculated by substituting the values of the parameters. The integration $\int_{R_{ab}}^{R_{ab}+2R_{roc}} \sin^{-1}\left(\frac{R_{ab}}{x}\right) dx$ in Eq.(4.33) does not have an analytic solution, and from Eq.(4.9) we know that R is a function of a and v, so we cannot get the explicit expression of T with respect to a and v. However, sometimes analytic solution is required to

analyze the effect of a and v. In this case, an efficient way is to find an approximate solution, which can be obtained by dealing $\sin^{-1}\left(\frac{R_{ab}}{x}\right)$ with Taylor expansion, according to the requirement of precision. If a function is infinitely derivative at the expansion point, it can be expanded by Taylor expansion.

The range of x in $\int_{R_{ab}}^{R_{ab}+2R_{roc}} \sin^{-1}\left(\frac{R_{ab}}{x}\right)dx$ is $[R_{ab}, R_{ab}+2R_{roc}]$. If there are no limits on the value of a and v, R can range from 0 to ∞. Regardless of the values of a and v, the range of $\frac{R_{ab}}{x}$ will be $(0,1)$. Furthermore, based on the analysis of error, the order of expansion and the point of expansion can be decided. For instance, we let $R_{ab} = 0.5$ in the subsequent analysis and simulation, and the values of a and v be controllable, so we select the expansion point at $x = 1$, which is believed to be reasonable. Because the function

$$f(x) = \sin^{-1}\left(\frac{R_{ab}}{x}\right) \tag{4.34}$$

is infinitely derivative at point $x = 1$, the function can be expanded at this point.

In mathematics, a function that satisfies some specific conditions can be presented as the sum of Taylor series. The Taylor series of a function $f(x)$ about point $x = a$ is

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x-a)^n \tag{4.35}$$

where n is a positive integer, and $n!$ is the factorial of n, $f^{(n)}(a)$ is the nth derivative of $f(x)$ at the point a. In the application of Taylor series, we always

calculate finite number of terms, which is the nth order Taylor polynomial

$$T_n(x) = f(a) + f'(s)(x-a) + \frac{f'(a)}{2!}(x-a)^2 + \frac{f''(a)}{3!}(x-a)^3 + \cdots + \frac{f^{(n)}(a)}{n!}(x-a)^n \quad (4.36)$$

The remainder is defined as the difference between Taylor series and the nth order Taylor polynomial

$$R_n(x) = f(x) - T_n(x) \quad (4.37)$$

Lagrange remainder term is one of the commonly used remainders, which is

$$R_n(x) = \frac{f^{(n+1)}(\mu)}{(n+1)!}(x-a)^{n+1} \quad (4.38)$$

where μ is in the neighbourhood of $x = a$. If the Taylor expansion converges at $f(x)$, the Lagrange remainder should be 0 when n approaches ∞

$$\lim_{n\to\infty} R_n(x) = 0 \quad (4.39)$$

Therefore, we only need to check Eq.(4.39) to prove the convergence of the Taylor expansion of Eq.(4.34). By calculation, the nth order derivative of $f(x)$ is of the same order to a function of R_{ab}/x to the $(n+1)$th power. According to the values of R_{ab} and a that we select, the convergence of $(\frac{R_{ab}}{x})^n \frac{1}{n!}$ can be known, so we can decide the convergence of Eq.(4.34).

In this study, we select Taylor expansion around the point $x = 1$, and the approximate results are compared with the original function of Eq.(4.34) by Figure 4.9 and Figure 4.10. The second-order Taylor expansion of Eq.(4.34) around the point $x = 1$ is:

$$f(x) = \frac{\pi}{6} - \frac{\sqrt{3}}{3}(x-1) + \frac{7\sqrt{3}}{18}(x-1)^2 + O((x-1)^3) \tag{4.40}$$

The Lagrange remainder term for the second order expansion is:

$$R_2(x) = \left[-\frac{1}{2\xi^4\left(1-\frac{1}{4\xi^2}\right)^{\frac{1}{2}}} - \frac{7}{48\xi^6\left(1-\frac{1}{4\xi^2}\right)^{\frac{3}{2}}} - \frac{1}{64\xi^8\left(1-\frac{1}{4\xi^2}\right)^{\frac{5}{2}}} \right] \cdot (x-1)^3,\ \xi \in (1,x)$$

(4.41)

The accuracy of approximation can be higher by increasing the order of expansion. We have also calculated the forth order Taylor expansion of Eq.(4.34) at point $x = 1$, which is:

$$f(x) = \frac{\pi}{6} - \frac{\sqrt{3}}{3}(x-1) + \frac{7\sqrt{3}}{18}(x-1)^2 - \frac{13\sqrt{3}}{27}(x-1)^3 + \frac{205\sqrt{3}}{324}(x-1)^4 + O((x-1)^5) \tag{4.42}$$

The Lagrange remainder term for the forth order expansion is:

$$R_4(x) = \left[-\frac{1}{2\xi^6\left(1-\frac{1}{4\xi^2}\right)^{\frac{1}{2}}} - \frac{3}{8\xi^8\left(1-\frac{1}{4\xi^2}\right)^{\frac{3}{2}}} - \frac{183}{1280\,\xi^{10}\left(1-\frac{1}{4\xi^2}\right)^{\frac{5}{2}}} - \frac{13}{512\,\xi^{12}\left(1-\frac{1}{4\xi^2}\right)^{\frac{7}{2}}} - \right.$$

$$\left. 74096\xi 141 - 14\xi 292 \cdot x - 15,\ \xi \in (1,x) \right] \tag{4.43}$$

In order to analyze the function more clearly, second order expansion will be used in the following simulation.

The integration $\int_{R_{ab}}^{R_{ab}+2R_{roc}} \sin^{-1}\left(\frac{R_{ab}}{x}\right) dx$ can be approximated by applying the second order expansion, which is:

$$\int_{R_{ab}}^{R_{ab}+2R_{roc}} \sin^{-1}\left(\frac{R_{ab}}{x}\right) dx \approx \frac{224\sqrt{3}}{27}\frac{v^6}{a^3} + \frac{52\sqrt{3}}{9}\frac{v^4}{a^2} + \left(\frac{2\pi}{3} + \frac{19\sqrt{3}}{18}\right)\frac{v^2}{a} \tag{4.44}$$

where the Eq.(4.9) is substituted into the function. Figure 4.9 shows the comparison of the original integration and the approximate integration of the 2nd-order Taylor expansion. One curve is the arithmetic solution of $\int_{R_{ab}}^{R_{ab}+x} \sin^{-1}\left(\frac{R_{ab}}{x}\right) dx$, and the other one is the integration of the Taylor series of Eq.(4.34). Figure 4.10 shows the comparison of the original integration and the 4th-order approximation. It can be seen that when the Taylor series is of higher order, the approximate integration curve will be closer to the original curve. Therefore, we can approximate the analytic solution and analyze the effect of a and v by using Taylor expansion. On the other hand, function T with respect to n can be obtained easily by numerical integration by setting a and v to be constants.

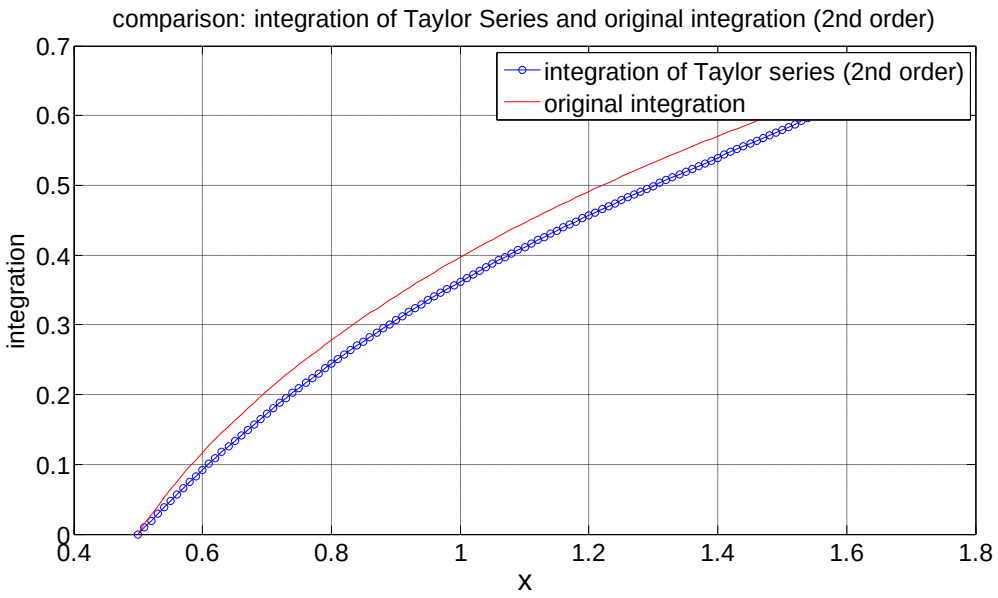


Figure 4.9 Comparism of 2nd-order Taylor expansion of the integration and the original integration.

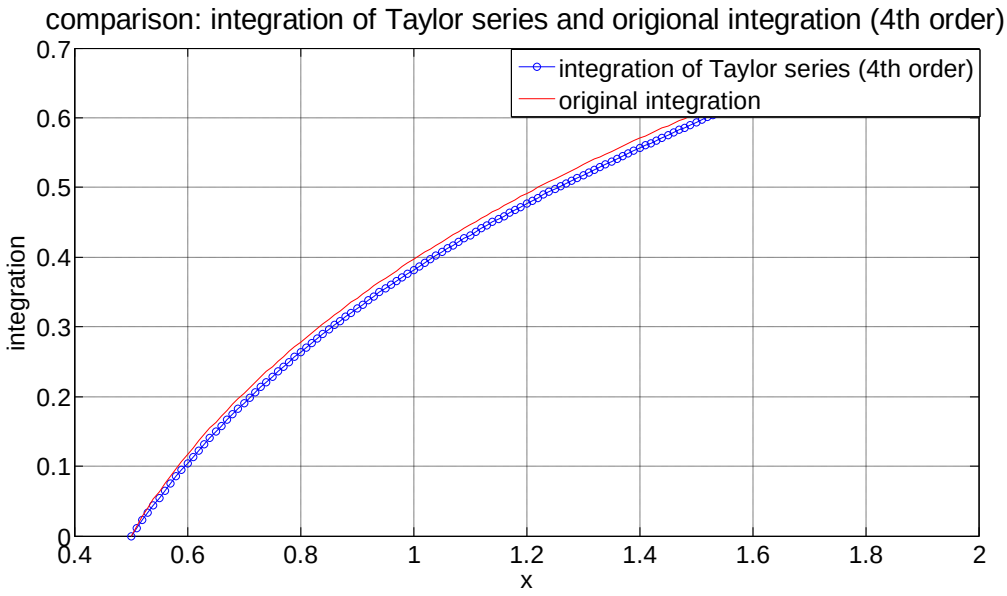


Figure 4.10 Comparism of 4th-order Taylor expansion of the integration and the original integration.

4.5.2 Critical Number of Vehicles

Since we have known the formula for the time to first collision, the critical number of vehicles can be derived. If we expect that the first collision in the confined area happens beyond some certain time, the range of variables and the relation between them can be known from the formula. When we let a and v be constants, T is a function of n, and T will increase when n decreases. In (4.33), we let

$$M_1 = C_1 \cdot \frac{S^2 a}{\sqrt{2} v (4v^2 + R_{ab}\, a) R_{ab} \int_{R_{ab}}^{R_{ab}+2R_{roc}} \sin^{-1}\left(\frac{R_{ab}}{x}\right) dx} \tag{4.45}$$

and

$$M_2 = C_2 \cdot \frac{S}{2\sqrt{2} R_{ab}\, v} \tag{4.46}$$

Suppose that a and v are fixed to constants, so Eq.(4.33) can be rewritten as

$$T = M_1 \cdot \frac{1}{(n-1)^2} + M_2 \cdot \frac{1}{n-1} \tag{4.47}$$

We denote the critical time as T_{cr}, which indicates that we want the first collision to happen beyond time T_{cr} since operating. We let the time be critical time, which is

$$T = T_{cr} \tag{4.48}$$

Therefore, by solving Eq.(4.48), the critical number of vehicles can be obtained

$$n_{cr} = 1 + \frac{M_2}{2T_{cr}} \pm \frac{\sqrt{4TM_1 + {M_2}^2}}{2T_{cr}} \tag{4.49}$$

Similar to that in the previous chapter, we have known the critical number of vehicles of the system, when the critical time is defined. As a result, if the number of vehicles is selected to be less than the critical number of vehicles n_{cr}, we can expect the time to first collision to be larger than the critical time T_{cr}. Similarly, when some of the variables in the formula are determined, we can deduce the proper values of the undetermined variables according to the requirement of the task.

4.6 Simulation and Discussion

This section provides the results of Monte Carlo simulation, and it will verify the formula established in the previous sections. Numerical integration is used in the formula when the speed and acceleration are fixed, while Taylor expansion is applied to simulate with respect to the speed and acceleration. The constants in the formula are approximated in the case where the size of

the workspace and that of vehicles are as specified in the simulation.

4.6.1 Parameters of Vehicles and Workspace

Consistent with the simulation of previous study, the workspace in the simulation is set to be a square of $20\text{m} \times 20\text{m}$, where the operation area is scaled to $x(\text{m}) \in [-10,10], y(\text{m}) \in [-10,10]$. The range of the workspace is proportional to that in the last chapter, so the result in this study is comparable with the one in the last chapter. Furthermore, the study can be made by changing the scale disproportionally.

In these simulation experiments, the initial positions and velocities of the vehicles are given randomly in the workspace, which are presented by coordinates in the coordinate frame. Once a set of variables a, v and n are selected, we can get the time to first collision from Monte Carlo simulation. In order to get a statistical result, 200 independent trials were done for each set of variables, and the mean values of the simulation results are obtained. When one of the three variables varies, the mean values can be fitted to the theoretical curve Eq.(4.33), so that the form of the formula with respect to the varying variable can be verified, and the constants in the formula can be approximated. When the distance between the centres of two vehicles is less than twice of their radius, or when the distance between one vehicle and the boundary of the workspace, collision is considered to occur. In this case, the program stops to record the time that it took from beginning.

In the simulations, the number of vehicles n varies from 5 to 50 (simulation step: 1). Speed v is from $0.5(\text{m/s})$ to $2(\text{m/s})$ (simulation step:

$0.1(\mathrm{m/s})$), and the acceleration a is from $1(\mathrm{m/s^2})$ to $5(\mathrm{m/s^2})$ (simulation step: $0.4(\mathrm{m/s^2})$), and the average time to first collision is recorded as the result. Besides, we suppose that the radius of the vehicle is $R_{ob} = 0.25\mathrm{m}$.

The process of moving can also be seen visually, and the interface is shown in Figure 4.11. The colorful dots in Figure 4.11 represent the vehicles, and this figure only shows the case where one vehicle is moving and the others are static for easy understanding. We can see from the visualized interface that the moving vehicle turns 90° (absolute value) when it come across an obstacle, according to different relative positions between the reference vehicle and obstacle vehicle, and the program stops when the collision with an obstacle vehicle cannot be avoided.

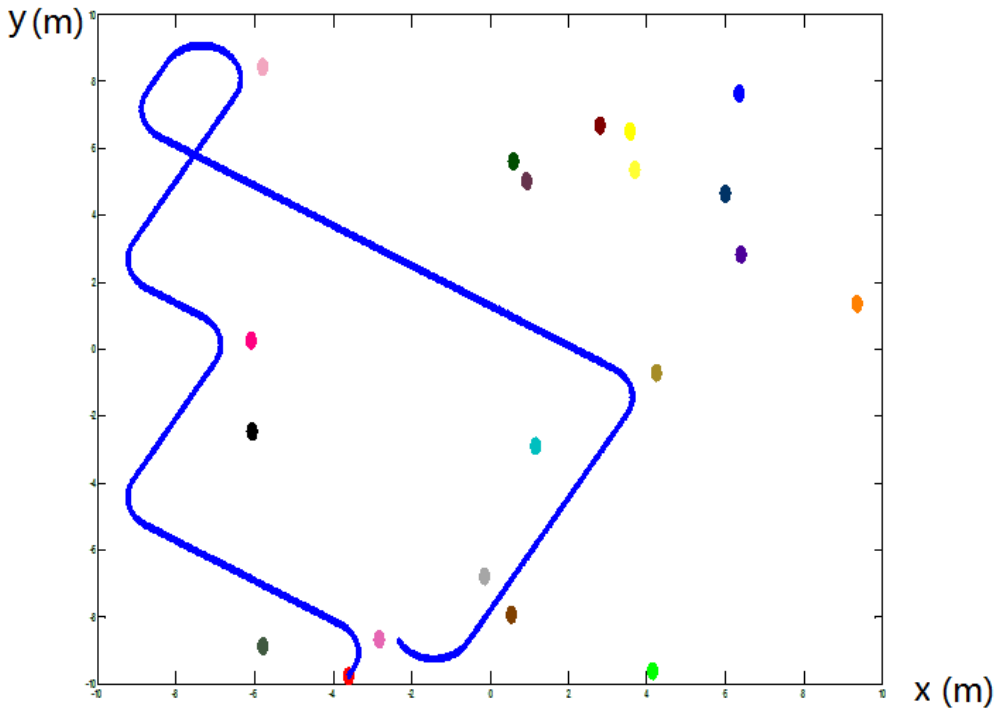


Figure 4.11 Visualized interface of the running program for Dubins' vehicles.

4.6.2 Flow Chart of Program

The flow chart of the program in this study is shown in Figure 4.12. The initial positions and velocities are generated randomly without overlapping. In each time step, collision between every two vehicles is checked. All of the vehicles are supposed to move straight forward at the beginning, and when some vehicle finds that it may collide with an obstacle, it will begin tuning motion. The program forwards one step after the collisions between every two vehicles and between vehicles and boundaries are checked. The program will stop when collision happens. This program is repeated 200 times to get statistical results for each set of combination of variables.

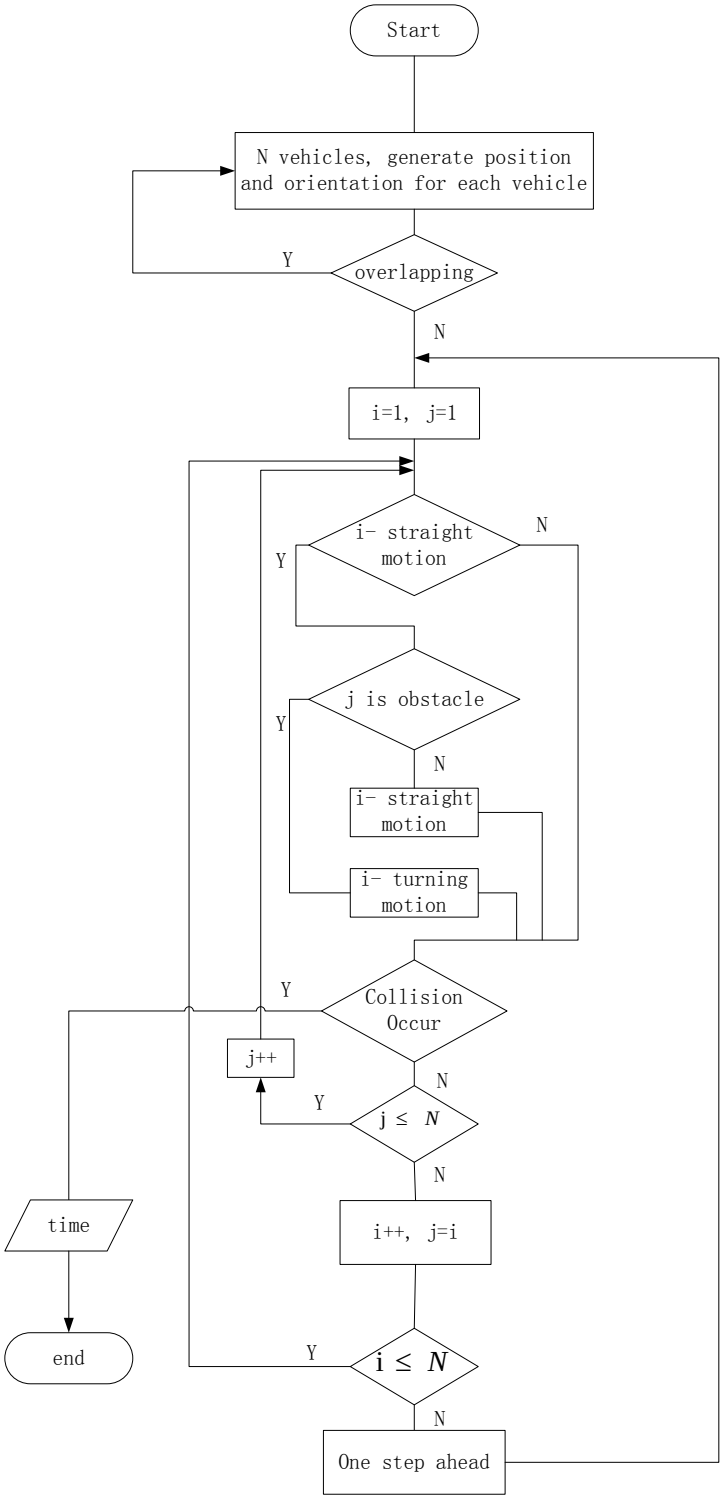


Figure 4.12 Flow graph of the program for calculating the time to first collision for Dubins' vehicles with non-zero turn radius in a confined area.

4.6.3 Simulation Results

The simulation results will be shown in this section, and will be fitted to the

theoretical curves to compare to the formula developed in previous sections. Formula Eq.(4.33) is a function of n, v and a, assuming that the sizes of workspace and vehicles are fixed. In order to verify if the simulation result agrees with Eq.(4.33), the mean values of simulation results are fitted to the curves in terms of each variable. First, numerical solutions of the integration term in Eq.(4.33) are calculated with different values of v and a, and the form of the formula with respect to n can be verified by fitting. The constants in the formula will be approximated for the case in the simulations, and the constants are substituted into the formula in the following fitting. Second, in order to check the form of formula with respect to v and a, Taylor series is applied to substitute for the integration form, and the simulation results are fitted subsequently. Besides, in order to see the goodness of fit of the fit curves, the coefficient of determination ${R_d}^2$ is also provided in this study, which is between 0 and 1. An ${R_d}^2$ value closer to 1 indicates that the curve has a better fit to the data.

In the following, the fitting results in terms of three variables will be shown and the results will be illustrated. Three examples with different variables are selected, and the effects of variables n, a and v on the expected time to first collision T are found respectively.

4.6.3.1 Effect of the Number of Vehicles

In this section, the form of formula Eq.(4.33) with respect to n is checked. While studying the effect of n, we let the other variables in formula Eq.(4.33) be constants. In other words, a and v are set to be constants. For every

combination of a and v, the arithmetic solution term in the formula will be calculated, and the formula is a function of n. Therefore, the time to first collision can be collected by Monte Carlo simulation for different ns. Furthermore, the average time to first collision will be fitted to the theoretical formula of n. An example is illustrated in Figure 4.13, which is the case where $v = 1\mathrm{m/s}, a = 2.6\mathrm{m/s}^2$. The number of vehicles ranges from 5 to 50, and the mean values of Ts from simulation for different ns are marked by "stars" in Figure 4.13 (a). The solid curve in the figure is the theoretical function calculated from formula Eq.(4.33) by substituting the values of a and v. Simulation data are fitted to the theoretical curve. In this example, the constants that are obtained from fitting are $C_1 = 106.6 \times 10^{-4}, C_2 = 10.5 \times 10^{-2}$, which can be approximate values of C_1 and C_2. Moreover, we also introduce the coefficient of determination ${R_d}^2$ to check the goodness of fit, and a value of ${R_d}^2$ closer to 1 indicates a better fitting. Here, the coefficient is ${R_d}^2 = 0.995$, which implies that the curve can fit the mean values of simulation data very well. Besides, the vertical lines on the points show the standard deviations of the data at these points. The standard deviation is not small, especially when the expected time is long. The same as that in Chapter 3, this is because the initial positions and the velocities of the vehicles are generated randomly, and they may have crucial influence on the time of the first collision.

It is clear from the figure that when the number of vehicles increases, the expected time to first collision will decrease. At the smaller densities of vehicles, the expected time of the first collision T changes rapidly with density,

and the change is relatively slow when T is large. In Figure 4.13 (b), the residues of fitting points are shown.

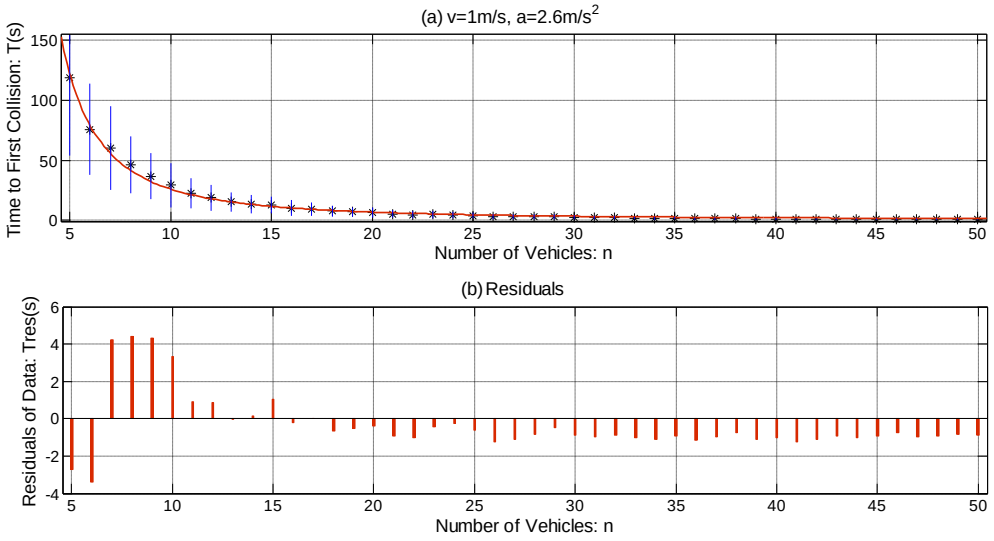


Figure 4.13 (a) Fit curve and residuals with respect to n when $v = 1\mathrm{m/s}, a = 2.6\mathrm{m/s}^2$. (b) residuals of the fitting curve.

In addition, the fitting results when a varies from 1 to 3.8 are shown in Figure 4.14. It can be seen that the simulation points can also fit the theoretical curve well. Comparison of these curves is shown in Figure 4.15, which indicates that when a becomes larger, the time to first collision is also larger, under the circumstance that v and n unchange. The reason is that a larger a indicates a smaller radius of curvature R_{roc}, and a vehicle is more possible to avoid collision with an obstacle with smaller radius of curvature R_{roc}.

The constants C_1 and C_2 in formula Eq.(4.33) can be obtained as well by fitting data for some selected accelerations, so that we can calculate the average values of them. The constants from the fitting curves are listed in Table 4.1, where $a = 1, 1.4, 1.8, 2.2, 2.6, 3, 3.4, 3.8, 4.2\mathrm{m/s}^2$ respectively. In Table 4.1, the constants C_1 and C_2 obtained from fitting are shown for different

values of a, and the coefficients of determination are also shown in the last column of the table. Therefore, we can approximate the values of C_1, C_2 and ${R_d}^2$, which are in the last row of the table, by averaging the values, and the expected values of them are $C_1 = 106.01 \times 10^{-4}$, $C_2 = 10.45 \times 10^{-2}$, ${R_d}^2 = 0.9531$. As expected by function Eq.(4.33), the relation between C_1 and C_2 should be $C_1 = {C_2}^2$, and it can be seen from Table 4.1 that the approximate values comfort to this relation generally.

Table 4.1 Constants obtained from simulations

$a(\mathrm{m/s^2})$	$C_1/10^{-4}$	$C_2/10^{-2}$	${R_d}^2$
1	109.20	10.88	0.9196
1.4	108.50	11.39	0.9023
1.8	110.80	10.18	0.9392
2.2	110.40	10.46	0.9777
2.6	106.60	10.50	0.9950
3	102.80	10.06	0.9742
3.4	106.20	10.42	0.9873
3.8	100.57	10.15	0.9288
4.2	99.06	10.02	0.9553
mean	106.01	10.45	0.9533

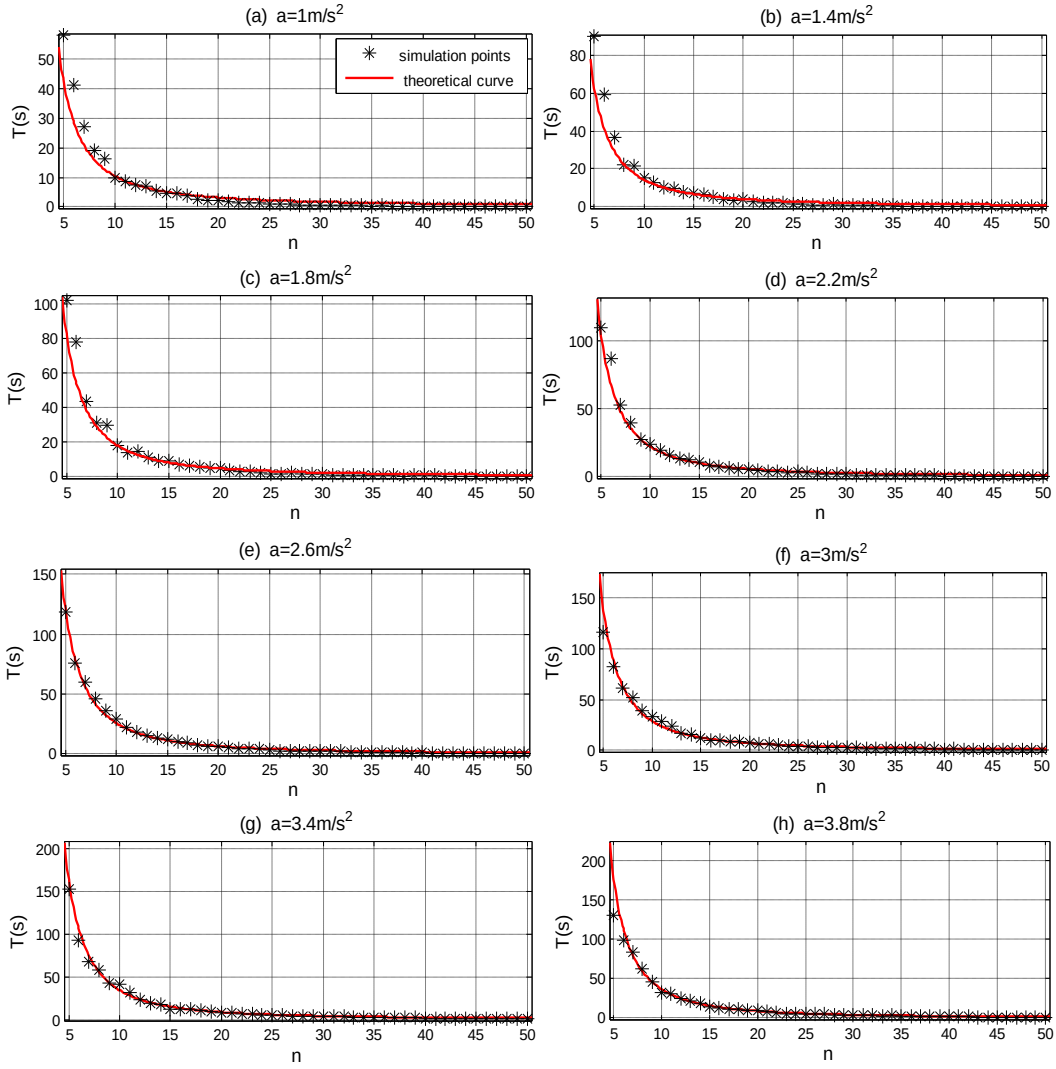


Figure 4.14 Effect of varying acceleration from $1\mathrm{m/s^2}$ to $3.8\mathrm{m/s^2}$.

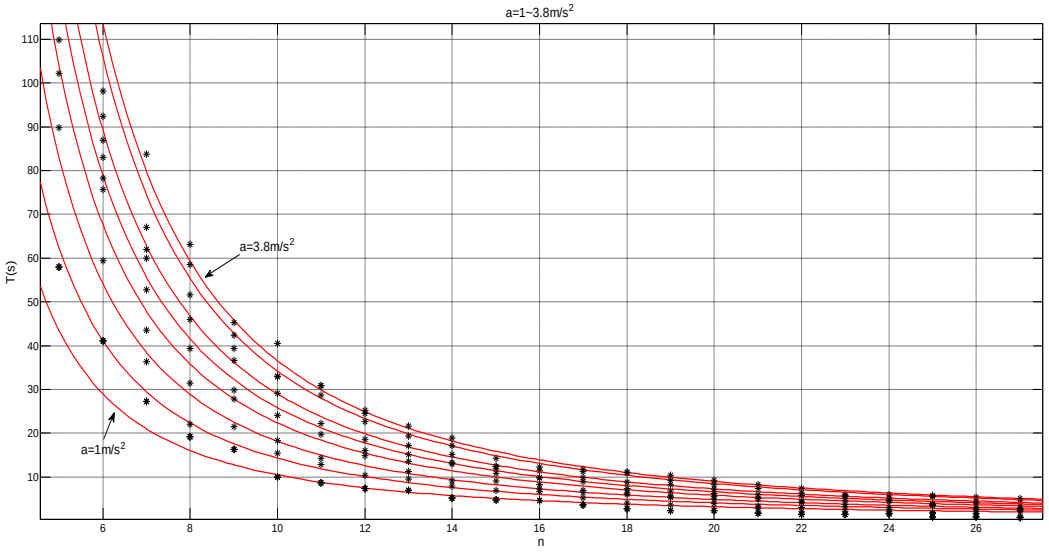


Figure 4.15 Data of Figure 4.14 presented in one plot.

4.6.3.2 Effect of Acceleration and Speed

Since the constants C_1 and C_2 in formula Eq.(4.33) have been approximated in the last section, we can substitute the approximated values back into the formula Eq.(4.33). We will check the form of the formula with respect to a and v directly by comparing the simulation values and the theoretical curve. Some examples will be shown here. Because the integration term in the formula includes variables a and v, we will use Taylor expansion to get the explicit expression. 2nd-order Taylor Series expansion Eq.(4.44) is used to approximate the integration $\int_{R_{ab}}^{R_{ab}+2R_{roc}} \sin^{-1}\left(\frac{R_{ab}}{x}\right) dx$ in formula Eq.(4.33).

Figure 4.16 shows the comparison of the theoretical $a-T$ curve and the mean values of simulation data, When $v = 1\text{m/s}, n = 22$. The constants C_1 and C_2 that are used are the approximate values above ($C_1 = 106.01 \times 10^{-4}$, $C_2 = 10.45 \times 10^{-2}$). The comparison shows that the coefficient of

determination in this case is ${R_d}^2 = 0.9917$, if the comparison is regarded as a curve fitting. Another example when $v = 1\mathrm{m/s}, n = 24$ is shown in Figure 4.17. The coefficient of determination in this case is ${R_d}^2 = 0.9804$.

We have also checked $v - T$ curve based on the approximated constants. In Figure 4.18, the solid curve is the theoretical $v - T$ curve when $a = 2.4\mathrm{m/s}^2, n = 24$, and the constants C_1 and C_2 are also the approximated ones. The coefficient of determination in Figure 4.18 is ${R_d}^2 = 0.9561$.

All of the curves have a relatively high coefficient of determination, so the coefficients that are obtained from last section are validated, and the form of formula Eq.(4.33) is verified in terms of variables a and v. In the figures, it is shown that the time to first collision increases, when a increases and v decreases.

The theoretical curves with respect to a and v we used in the simulations are just approximated, which is to check the form of the formula. If more accurate results are wanted, the integration in formula Eq.(4.33) can be expanded to a higher order. However, the expression of higher order expansion will be more complicated, and more computationally expensive.

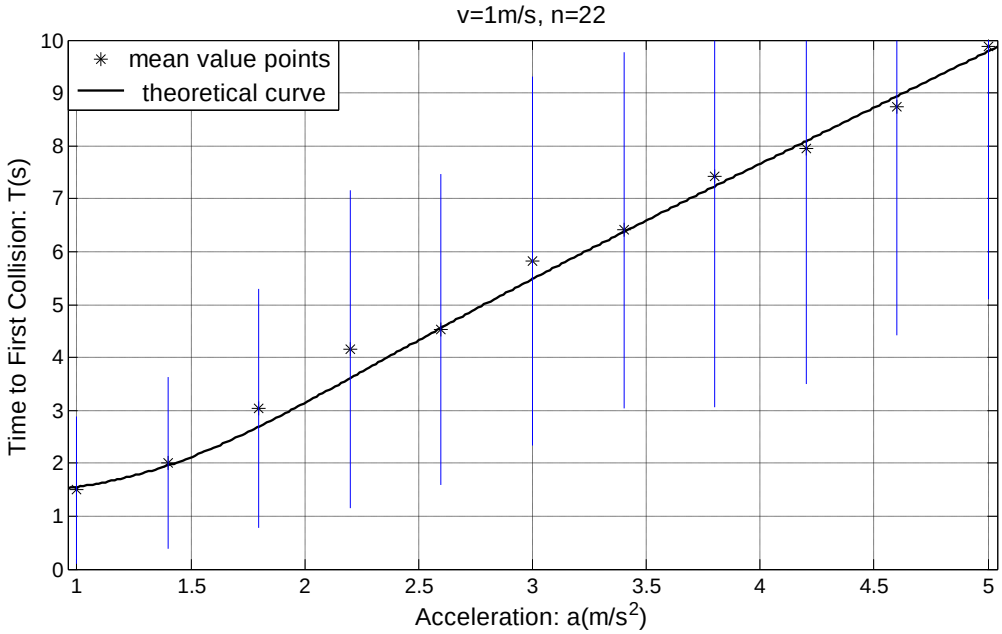


Figure 4.16 Fit Comparison between theoretical curve and mean value points from simulation when v=1m/s, n=22.

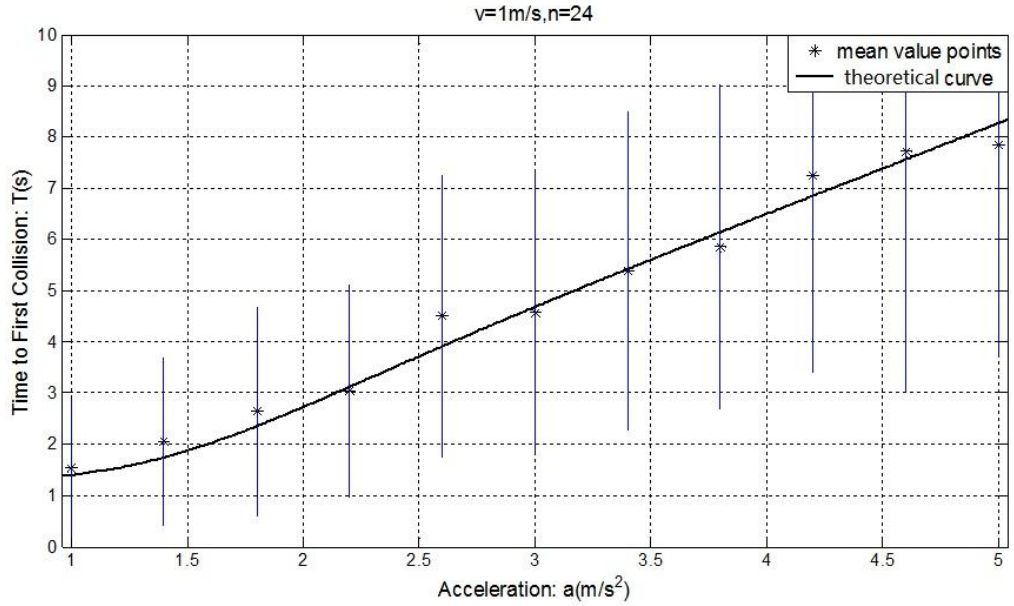


Figure 4.17 Comparison between theoretical curve and mean value points from simulation when v=1m/s, n=24.

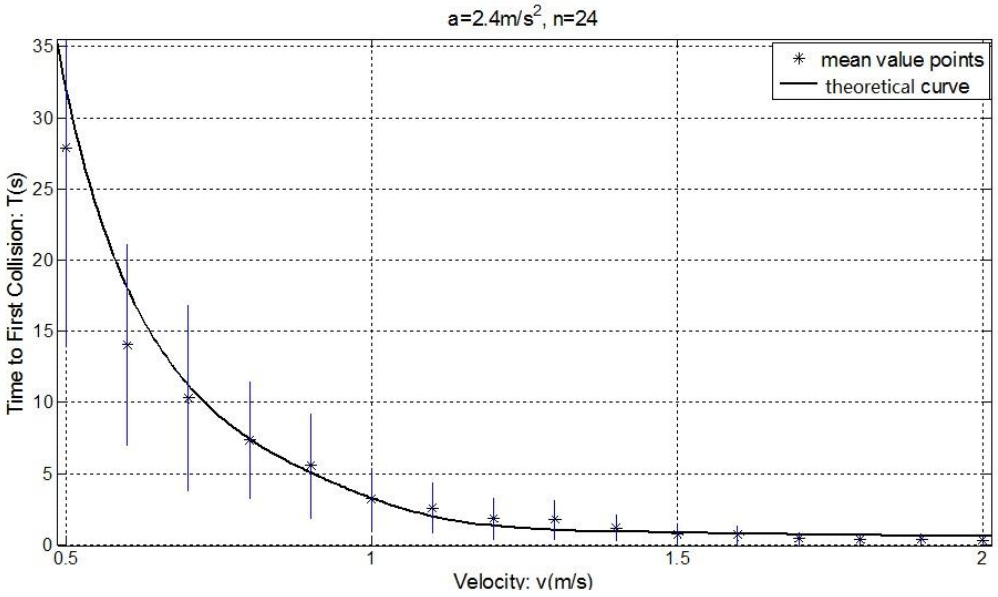


Figure 4.18 Comparison between theoretical curve and mean value points from simulation when a=2.4 m/s^2, n=24.

4.7 Conclusion

In this chapter, we explored the time to first collision when a swarm of Dubins' vehicles operate in a confined area, while Velocity Obstacle is applied as the collision avoidance approach. The model of Dubins' vehicle was adopted, and all the vehicles move in the same pattern. The initial positions and velocities of the vehicles are generated randomly, and the vehicles avoid collision with each other using Velocity Obstacle method. In this study, the time to first collision T is derived referring to the derivation of the Mean Free Path in molecular dynamics. In a confined area S, T is a function of the speed v, acceleration a and the number of vehicles n, if the vehicles have fixed sizes. There is an integration term in the formula of T, so the explicit expression of the function was approximated by second-order Taylor Series. Moreover, Monte Carlo simulations were done to validate the formula, and the constants in the formula were approximated from the simulations and validated. From

both theoretical formula and simulation results, it can be seen that the time to first collision increases, when a increases or when v and n decrease. As shown by the formula, in case that the time to first collision, which is required by the task, is given in motion planning, the range of variables can be calculated. In other words, the combination of the number of vehicles, speed and acceleration can be found to fulfill the requirement of the task. Although the study in this paper is based on Dubins' vehicle and Velocity Obstacle method, the idea can also be applied into studies on other vehicle models or collision avoidance methods, and path planning of a swarm of unmanned vehicles in a confined area.

Chapter 5. Conclusions and Future Works

5.1 Conclusions

The main objective of this thesis is to find the time to first collision of multiple UxVs in a confined area. The studies were conducted on two kinds of nonholonomic vehicles and two kinds of collision avoidance methods. In each case, the time to first collision in a confined area is presented as a function of the parameters of the dynamic models of UxVs and the environments. The influence of each parameter is quantified by formulas and analyzed. The critical number of UxVs was derived, and when the number of vehicles is below the critical number, the time to first collision can be expected to exceed the critical time. Besides, the effect of the boundaries of operation area was also considered in studies.

In Chapter 3, the time to first collision of UxVs with zero turn radius in a confined area was derived. First, the derivation of mean free path for molecules in an open area was introduced, which is used in this thesis. The derivation of the time to first collision for circular unmanned vehicles in an open area was also shown as a basis for the following studies. For the study in terms of UxVs with zero turn radius, a vehicle model with limited field of view (FOV), constant speed, zero turn radius was proposed. When the sizes of the vehicles and the operation area are fixed, the expected time to first collision is a function of the number of vehicles n, speed v and FOV φ. The time to first collision indicates the probability of collision of the UxVs within

the area. The influences of the parameters can be seen from the formula Eq.(3.19). If the number of vehicles $n = 1$, T will be infinite. This means that when there is only one vehicle in the area, it will never collide with other vehicles. If the speed of vehicle $v = 0$, T will also be infinite, which indicates that the vehicle has no speed. Larger FOV may also result in larger T. Consequently, the expected time to first collision will decrease at a decreasing rate while the number of vehicles and the velocity of vehicles increase, but it will increase at an increasing rate when FOV increases. Furthermore, the formula of critical number of UxVs was derived. The critical number can be determined when the critical time is defined, below which collision can be deemed to occur instantaneously. Monte Carlo simulations were conducted to verify the formula developed. Two of the three variables n, v and φ were set to constants, the expected time to first collision and standard deviations in each case were obtained. The simulation data were fitted to the curve that is derived mathematically, and the goodness of fit was checked. From the simulation and fitting results, we can see that the coefficients of determination, a value in statistics that indicates how well data fit a statistical model, are close to 1, which indicates a good fit between the theoretical formula and the simulation points. Besides, the values of the coefficients in the formula were approximated from the simulations to calculate the critical number of UxVs. Consequently, the theory developed was validated by the simulations.

In Chapter 4, we studied the time to first collision for Dubins' vehicles with non-zero Turn Radius in a confined Area. Velocity Obstacle method is applied as the collision avoidance method. All the vehicles move in the same

pattern, and the initial positions and velocities of the vehicles are generated randomly. The idea from the derivation of the Mean Free Path in molecular dynamics is also used in this study. For fixed sizes of vehicles and operation area, the time to first collision is derived as a formula with respect to the speed v, acceleration a and the number of vehicles n. The relations among different variables are revealed by the formula, and the formula for critical number of vehicles was also derived. There is an integration term in the formula, so this term was approximated by second-order Taylor Series in case that explicit expression with respect to the variables is required. Monte Carlo simulations were also done in this study to validate the formula. We also set two of the three variables to be constants, and the points of the expected time to first collision from simulations were fitted to the curve from the formula. The coefficients of determination are close to 1, so it can be seen that the fittings are very well, and the form of the formula is verified. From both theoretical formula and simulation results, it is revealed that the time to first collision increases, when a increases or when v and n decrease. Besides, the constants in the formula were approximated from the simulations and validated.

As a consequence, the time to first collision in two different cases were derived, and verified by Monte Carlo simulations. The values of system parameters can then be selected especially the number of vehicles, according to the formulas to fulfill task requirements before actual operation. For example, when some of the parameters of the vehicles, such as speed, acceleration, are fixed, the maximum number of vehicles that can operate in the area can be derived according to the formula in this study. Therefore, the results in this thesis are useful as references for UxVs operations in confined areas. The

results allow us to make a plan on the system parameters of a swarm of UxVs in a confined area. If the model of the vehicle can be changed for different situations, the variables affecting collision will be different. However, although the study in this thesis is based on two specific vehicle models and collision avoidance methods, the idea can also be applied into studies on other vehicle models or collision avoidance methods according to the ideas in this thesis. When the model of vehicles is built, the time to first collision can be found based on the motion law of vehicles.

5.2 Limitations and Future works

It should be noted that there are some limitations for the studies in this thesis. Several limitations are listed below, and the recommendations for future works are given.

Firstly, in this thesis, it was assumed that all the UxVs in the confined area have the same motion patterns, speeds and accelerations. It is more convenient to see the properties of some specific vehicle model and collision avoidance method, when the vehicles move in the same pattern. However, in order to deal with more complicated situations, the cases where the vehicles move with different motion patterns are also of interest. The assumption can be relaxed and different models can be investigated in future.

Secondly, the UxVs in this thesis only change the orientation of velocities while avoiding collisions. The work can be extended to other kinds of collision avoidance methods, for example changing the magnitude of the speed.

Thirdly, for the studies in this thesis, the UxVs move freely in the operation area without destination, and they only stop when collision happens. To address this problem, future research can attempt to conduct on UxVs with respective destinations, so that collision in multi-UxV tasks can be studied.

Moreover, only simulation results were used to verify the theories developed in this thesis. In future, more experiments may be conducted to improve the studies.

Bibliography

[1] L. Hao and F. Nashashibi, "Cooperative multi-vehicle localization using split covariance intersection filter," in *Intelligent Vehicles Symposium (IV), 2012 IEEE*, 2012, pp. 211-216.

[2] A. R. Girard, J. B. de Sousa, and J. K. Hedrick, "An overview of emerging results in networked multi-vehicle systems," in *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on*, 2001, pp. 1485-1490 vol.2.

[3] A. Gautam and S. Mohan, "A review of research in multi-robot systems," in *Industrial and Information Systems (ICIIS), 2012 7th IEEE International Conference on*, 2012, pp. 1-5.

[4] *Intelligent System Control Integration Laboratory*. Available: http://isci.cn.nctu.edu.tw

[5] *MATLAB*.Available: http://www.mathworks.com/products/matlab/features.html#data_analysis

[6] P. Wilke and T. Bräunl, "Flexible wireless communication network for mobile robot agents," *Industrial Robot: An International Journal*, vol. 28, pp. 220-232, 2001.

[7] L. Steels, "Evolving grounded communication for robots," *Trends in Cognitive Sciences*, vol. 7, pp. 308-312, 2003.

[8] T. Balch and R. Arkin, "Communication in reactive multiagent robotic systems," *Autonomous Robots*, vol. 1, pp. 27-52, 1994/03/01 1994.

[9] F. Yus, "Misunderstandings and explicit/implicit communication," *Pragmatics,* vol. 9, pp. 487-518, 1999.

[10] K. I. Easton and A. Martinoli, "Efficiency and optimization of explicit and implicit communication schemes in collaborative robotics experiments," in *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, 2002, pp. 2795-2800.

[11] M. McPartland, S. Nolfi, and H. A. Abbass, "Emergence of communication in competitive multi-agent systems: a pareto multi-objective approach," in *Proceedings of the 2005 conference on Genetic and evolutionary computation*, 2005, pp. 51-58.

[12] A. Mutazono, M. Sugano, and M. Murata, "Frog call-inspired self-organizing anti-phase synchronization for wireless sensor networks," in *Nonlinear Dynamics and Synchronization, 2009. INDS'09. 2nd International Workshop on*, 2009, pp. 81-88.

[13] D. Sutantyo and P. Levi, "Decentralized underwater multi-robot communication using bio-inspired approaches," *Artificial Life and Robotics,* pp. 1-7, 2015/03/13 2015.

[14] M. Novitzky, C. Pippin, T. R. Collins, T. R. Balch, and M. E. West, "Bio-inspired multi-robot communication through behavior recognition," in *Robotics and Biomimetics (ROBIO), 2012 IEEE International Conference on*, 2012, pp. 771-776.

[15] C. Tatkeu, P. Deloof, Y. Elhillali, A. Rivenq, and J. M. Rouvaen, "A cooperative radar system for collision avoidance and communications between

vehicles," in *Intelligent Transportation Systems Conference, 2006. ITSC '06. IEEE*, 2006, pp. 1012-1016.

[16] T. Wang, Q. Dang, and P. Pan, "A Multi-Robot System Based on A Hybrid Communication Approach," *Studies in Media and Communication*, vol. 1, pp. 91-100, 2013.

[17] Y. Yamauchi and M. Yamashita, "Pattern formation by mobile robots with limited visibility," in *Structural Information and Communication Complexity*, ed: Springer, 2013, pp. 201-212.

[18] H. Yamaguchi, T. Arai, and G. Beni, "A distributed control scheme for multiple robotic vehicles to make group formations," *Robotics and Autonomous Systems*, vol. 36, pp. 125-147, 2001.

[19] L. E. Parker, K. Fregene, Y. Guo, and R. Madhavan, "Multi-robot Localization, Mapping, and Path Planning," in *Multi-Robot Systems: From Swarms to Intelligent Automata: Proceedings from the 2002 NRL Workshop on Multi-Robot Systems*, 2013, p. 21.

[20] E. Bahceci, O. Soysal, and E. Sahin, "A review: Pattern formation and adaptation in multi-robot systems," *Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-03-43*, 2003.

[21] A. S. Brandao, J. Barbosa, V. Mendoza, M. Sarcinelli-Filho, and R. Carelli, "A multi-layer control scheme for a centralized UAV formation," in *Unmanned Aircraft Systems (ICUAS), 2014 International Conference on*, 2014, pp. 1181-1187.

[22] G. Antonelli, F. Arrichiello, F. Caccavale, and A. Marino,

"Decentralized centroid and formation control for multi-robot systems," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, 2013, pp. 3511-3516.

[23] X. Li, M. F. Ercan, Y. Zhou, and F. Yu-Fai, "Algorithm for swarm robot flocking behavior," in *Autonomous Robots and Agents, 2009. ICARA 2009. 4th International Conference on*, 2009, pp. 161-165.

[24] Y. Mao, L. Chengfeng, and T. Yantao, "Flocking for Swarm Robot System: Distributed Coadaptive Control and Optimization," in *Information Engineering and Computer Science, 2009. ICIECS 2009. International Conference on*, 2009, pp. 1-4.

[25] A. V. Savkin, "Coordinated collective motion of groups of autonomous mobile robots: Analysis of Vicsek's model," *IEEE Transactions on Automatic Control*, vol. 49, pp. 981-983, 2004.

[26] Z. Li and J. Canny, *Nonholonomic motion planning*: Springer Science & Business Media, 1993.

[27] T. Liu and Z.-P. Jiang, "Distributed formation control of nonholonomic mobile robots without global position measurements," *Automatica,* vol. 49, pp. 592-600, 2013.

[28] D. Yoshida, T. Masuzawa, and H. Fujiwara, "Fault-tolerant distributed algorithms for autonomous mobile robots with crash faults," *Systems and computers in Japan*, vol. 28, pp. 33-43, 1997.

[29] W. Ren and N. Sorensen, "Distributed coordination architecture for multi-robot formation control," *Robotics and Autonomous Systems,* vol. 56, pp.

324-333, 2008.

[30] S. Souissi, Y. Yang, and X. Défago, "Fault-Tolerant Flocking in a k-Bounded Asynchronous System," in *Principles of Distributed Systems*. vol. 5401, T. Baker, A. Bui, and S. Tixeuil, Eds., ed: Springer Berlin Heidelberg, 2008, pp. 145-163.

[31] A. Guillet, R. Lenain, B. Thuilot, and P. Martinet, "Adaptable Robot Formation Control," 2014.

[32] R. Simmons, D. Apfelbaum, W. Burgard, D. Fox, M. Moors, S. Thrun, and H. Younes, "Coordination for multi-robot exploration and mapping," in *Proceedings of the National conference on Artificial Intelligence*, 2000, pp. 852-858.

[33] J. W. Fenwick, P. M. Newman, and J. J. Leonard, "Cooperative concurrent mapping and localization," in *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, 2002, pp. 1810-1817 vol.2.

[34] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *Robotics and Automation, IEEE Transactions on*, vol. 17, pp. 229-241, 2001.

[35] M. D. P. Moratuwage, W. S. Wijesoma, B. Kalyan, N. M. Patrikalakis, and P. Moghadam, "Collaborative multi-vehicle localization and mapping in high clutter environments," in *Control Automation Robotics & Vision (ICARCV), 2010 11th International Conference on*, 2010, pp. 1422-1427.

[36] T. Zhang, Z. Chong, B. Qin, J. Fu, S. Pendleton, and M. Ang, "Sensor fusion for localization, mapping and navigation in an indoor environment," in *Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM), 2014 International Conference on*, 2014, pp. 1-6.

[37] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, pp. 1052-1067, 2007.

[38] K. Konolige and M. Agrawal, "FrameSLAM: From bundle adjustment to real-time visual mapping," *Robotics, IEEE Transactions on*, vol. 24, pp. 1066-1077, 2008.

[39] H. Andreasson, T. Duckett, and A. Lilienthal, "Mini-SLAM: Minimalistic visual SLAM in large-scale environments based on a new interpretation of image similarity," in *Robotics and Automation, 2007 IEEE International Conference on*, 2007, pp. 4096-4101.

[40] M. Milford and A. George, "Featureless Visual Processing for SLAM in Changing Outdoor Environments," in *Field and Service Robotics*. vol. 92, K. Yoshida and S. Tadokoro, Eds., ed: Springer Berlin Heidelberg, 2014, pp. 569-583.

[41] L. Hao and F. Nashashibi, "Multi-vehicle cooperative localization using indirect vehicle-to-vehicle relative pose estimation," in *Vehicular Electronics and Safety (ICVES), 2012 IEEE International Conference on*, 2012, pp. 267-272.

[42] R. Irvine, "A geometrical approach to conflict probability estimation," *Air Traffic Control Quarterly,* vol. 10, pp. 85-113, 2002.

[43] M. Prandini, J. Hu, J. Lygeros, and S. Sastry, "A probabilistic approach to aircraft conflict detection," *Intelligent Transportation Systems, IEEE Transactions on,* vol. 1, pp. 199-220, 2000.

[44] J. Shen, N. H. McClamroch, and E. G. Gilbert, "A computational approach to conflict detection problems for air traffic control," in *American Control Conference, 1999. Proceedings of the 1999*, 1999, pp. 1445-1449 vol.2.

[45] M. Brannstrom, E. Coelingh, and J. Sjoberg, "Model-Based Threat Assessment for Avoiding Arbitrary Vehicle Collisions," *Intelligent Transportation Systems, IEEE Transactions on,* vol. 11, pp. 658-669, 2010.

[46] H. Blom, G. Bakker, M. Everdij, and M. Van der Park, "Collision risk modeling of air traffic," in *Proceedings of European Control Conference*, 2003.

[47] N. E. Du Toit and J. W. Burdick, "Probabilistic Collision Checking With Chance Constraints," *Robotics, IEEE Transactions on*, vol. 27, pp. 809-815, 2011.

[48] A. Soriano, E. Bernabeu, A. Valera, and M. Vallés, "Multi-Agent Systems Platform for Mobile Robots Collision Avoidance," in *Advances on Practical Applications of Agents and Multi-Agent Systems*. vol. 7879, Y. Demazeau, T. Ishida, J. Corchado, and J. Bajo, Eds., ed: Springer Berlin Heidelberg, 2013, pp. 320-323.

[49] F. Belkhouche, "Modeling and calculating the collision risk for air

vehicles," *Vehicular Technology, IEEE Transactions on,* vol. PP, pp. 1-1, 2013.

[50] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *The International Journal of Robotics Research,* vol. 17, pp. 760-772, 1998.

[51] A. Kimmel, A. Dobson, and K. Bekris, "Maintaining team coherence under the velocity obstacle framework," in *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, 2012, pp. 247-256.

[52] J. van den Berg, L. Ming, and D. Manocha, "Reciprocal Velocity Obstacles for real-time multi-agent navigation," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, 2008, pp. 1928-1935.

[53] J. van den Berg, J. Snape, S. J. Guy, and D. Manocha, "Reciprocal collision avoidance with acceleration-velocity obstacles," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 2011, pp. 3475-3482.

[54] J. K. Kuchar and L. C. Yang, "A review of conflict detection and resolution modeling methods," *Intelligent Transportation Systems, IEEE Transactions on,* vol. 1, pp. 179-189, 2000.

[55] Z. Xunyu, P. Xiafu, and Z. Jiehua, "Dynamic collision avoidance of mobile robot based on velocity obstacles," in *Transportation, Mechanical, and Electrical Engineering (TMEE), 2011 International Conference on*, 2011, pp. 2410-2413.

[56] J. Berg, S. Guy, M. Lin, and D. Manocha, "Reciprocal n-Body

Collision Avoidance," in *Robotics Research*. vol. 70, C. Pradalier, R. Siegwart, and G. Hirzinger, Eds., ed: Springer Berlin Heidelberg, 2011, pp. 3-19.

[57] S. Kumar, T. P. Parekh, and K. M. Krishna, "A hierarchical multi robotic collision avoidance scheme through robot formations," in *Robotics and Biomimetics (ROBIO), 2010 IEEE International Conference on*, 2010, pp. 306-311.

[58] D. Alejo, J. M. Díaz-Báñez, J. A. Cobano, P. Pérez-Lantero, and A. Ollero, "The Velocity Assignment Problem for Conflict Resolution with Multiple Aerial Vehicles Sharing Airspace," *Journal of Intelligent & Robotic Systems*, vol. 69, pp. 331-346, 2013/01/01 2013.

[59] S. A. Reveliotis and E. Roszkowska, "Conflict Resolution in Free-Ranging Multivehicle Systems: A Resource Allocation Paradigm," *Robotics, IEEE Transactions on*, vol. 27, pp. 283-296, 2011.

[60] R. A. Knepper and D. Rus, "Pedestrian-inspired sampling-based multi-robot collision avoidance," in *RO-MAN, 2012 IEEE*, 2012, pp. 94-100.

[61] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, 1985, pp. 500-505.

[62] S. Mastellone, D. M. Stipanović, C. R. Graunke, K. A. Intlekofer, and M. W. Spong, "Formation control and collision avoidance for multi-agent non-holonomic systems: Theory and experiments," *The International Journal of Robotics Research*, vol. 27, pp. 107-126, 2008.

[63] R. Conde, D. Alejo, J. Cobano, A. Viguria, and A. Ollero, "Conflict

Detection and Resolution Method for Cooperating Unmanned Aerial Vehicles," *Journal of Intelligent & Robotic Systems*, vol. 65, pp. 495-505, 2012/01/01 2012.

[64] M. Hoy, A. S. Matveev, and A. V. Savkin, "Algorithms for collision-free navigation of mobile robots in complex cluttered environments: a survey," *Robotica*, vol. 33, pp. 463-497, 2015.

[65] M. Hoy, A. S. Matveev, and A. V. Savkin, "Collision free cooperative navigation of multiple wheeled robots in unknown cluttered environments," *Robotics and Autonomous Systems*, vol. 60, pp. 1253-1266, 2012.

[66] Q. Huang, H. Ma, and H. Zhang, "Collision-avoidance mechanism of multi agent system," in *Robotics, Intelligent Systems and Signal Processing, 2003. Proceedings. 2003 IEEE International Conference on*, 2003, pp. 1036-1040 vol.2.

[67] D. V. Dimarogonas, S. G. Loizou, K. J. Kyriakopoulos, and M. M. Zavlanos, "A feedback stabilization and collision avoidance scheme for multiple independent non-point agents," *Automatica*, vol. 42, pp. 229-243, 2006.

[68] C. Chengtao, Y. Chunsheng, Z. Qidan, and L. Yanhua, "Collision Avoidance in Multi-Robot Systems," in *Mechatronics and Automation, 2007. ICMA 2007. International Conference on*, 2007, pp. 2795-2800.

[69] I. Škrjanc and G. Klančar, "Optimal cooperative collision avoidance between multiple robots based on Bernstein–Bézier curves," *Robotics and Autonomous Systems*, vol. 58, pp. 1-9, 2010.

[70] L. Yongwoo and K. Youdan, "Distributed unmanned aircraft collision avoidance using limit cycle," in *Control, Automation and Systems (ICCAS), 2011 11th International Conference on*, 2011, pp. 121-125.

[71] A. Rocha-Rocha, E. Munoz de Cote, S. P. Hernandez, and E. S. Succar, "Conflict Resolution in Multiagent Systems: Balancing Optimality and Learning Speed," in *Artificial Intelligence (MICAI), 2012 11th Mexican International Conference on*, 2012, pp. 32-37.

[72] D. Zhuoning, C. Zongji, Z. Rui, and Z. Rulin, "A hybrid approach of virtual force and A * search algorithm for UAV path re-planning," in *Industrial Electronics and Applications (ICIEA), 2011 6th IEEE Conference on*, 2011, pp. 1140-1145.

[73] D. Bodhale, N. Afzulpurkar, and N. T. Thanh, "Path planning for a mobile robot in a dynamic environment," in *Robotics and Biomimetics, 2008. ROBIO 2008. IEEE International Conference on*, 2009, pp. 2115-2120.

[74] Q. Yao-Hong, P. Quan, and Y. Jian-guo, "Flight path planning of UAV based on heuristically search and genetic algorithms," in *Industrial Electronics Society, 2005. IECON 2005. 31st Annual Conference of IEEE*, 2005, p. 5 pp.

[75] A. R. Willms and S. X. Yang, "An efficient dynamic system for real-time robot-path planning," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 36, pp. 755-766, 2006.

[76] G. C. Chasparis and J. S. Shamma, "Linear-programming-based multi-vehicle path planning with adversaries," in *American Control Conference, 2005. Proceedings of the 2005*, 2005, pp. 1072-1077 vol. 2.

[77] G. Lei, M.-z. Dong, T. Xu, and L. Wang, "Multi-Agent Path Planning for Unmanned Aerial Vehicle Based on Threats Analysis," in *Intelligent Systems and Applications (ISA), 2011 3rd International Workshop on*, 2011, pp. 1-4.

[78] D.-q. Zhang, J.-f. Zhao, M.-h. Wang, and G.-h. Niu, "Grey evaluation and optimization of UAV's path planning method," in *Electronic Computer Technology (ICECT), 2010 International Conference on*, 2010, pp. 85-88.

[79] H. Joo Young, K. Jun Song, L. Sang Seok, and P. Kyu-Ho, "A fast path planning by path graph optimization," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on,* vol. 33, pp. 121-129, 2003.

[80] Y. Mohan and S. Ponnambalam, "An extensive review of research in swarm robotics," in *Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on*, 2009, pp. 140-145.

[81] E. Şahin, "Swarm Robotics: From Sources of Inspiration to Domains of Application," in *Swarm Robotics*. vol. 3342, E. Şahin and W. Spears, Eds., ed: Springer Berlin Heidelberg, 2005, pp. 10-20.

[82] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, "Swarm robotics: a review from the swarm engineering perspective," *Swarm Intelligence,* vol. 7, pp. 1-41, 2013/03/01 2013.

[83] G. Beni, "The concept of cellular robotic system," in *Intelligent Control, 1988. Proceedings., IEEE International Symposium on*, 1988, pp. 57-62.

[84] G. Beni, "From Swarm Intelligence to Swarm Robotics," in *Swarm*

Robotics. vol. 3342, E. Şahin and W. Spears, Eds., ed: Springer Berlin Heidelberg, 2005, pp. 1-9.

[85] Y. Altshuler, V. Yanovsky, I. A. Wagner, and A. M. Bruckstein, "Efficient cooperative search of smart targets using UAV Swarms," *Robotica,* vol. 26, pp. 551-557, 2008.

[86] B. Walter, A. Sannier, D. Reiners, and J. Oliver, "UAV Swarm Control: Calculating Digital Pheromone Fields with the GPU," *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology,* vol. 3, pp. 167-176, July 1, 2006 2006.

[87] S. John, M. Robert, R. Joshua, M. John, and R. Stephanie, "Swarming Unmanned Air and Ground Systems for Surveillance and Base Protection," in *AIAA Infotech@Aerospace Conference*, ed: American Institute of Aeronautics and Astronautics, 2009.

[88] C. Robert, S. David, N. Todd, S. Witwicki, and B. Robert, "Cooperating Unmanned Vehicles," in *AIAA 1st Intelligent Systems Technical Conference*, ed: American Institute of Aeronautics and Astronautics, 2004.

[89] F. S. Schill, "Distributed communication in swarms of autonomous underwater vehicles," The Australian National University, 2007.

[90] A. Martinoli, K. Easton, and W. Agassounon, "Modeling swarm robotic systems: A case study in collaborative distributed manipulation," *The International Journal of Robotics Research,* vol. 23, pp. 415-436, 2004.

[91] O. Soysal and E. Sahin, "Probabilistic aggregation strategies in swarm robotic systems," in *Swarm Intelligence Symposium, 2005. SIS 2005.*

Proceedings 2005 IEEE, 2005, pp. 325-332.

[92] W. M. Spears, D. F. Spears, J. C. Hamann, and R. Heil, "Distributed, physics-based control of swarms of vehicles," *Autonomous Robots*, vol. 17, pp. 137-162, 2004.

[93] G. Francesca, M. Brambilla, A. Brutschy, V. Trianni, and M. Birattari, "AutoMoDe: A novel approach to the automatic design of control software for robot swarms," *Swarm Intelligence*, vol. 8, pp. 89-112, 2014.

[94] V. Trianni and S. Nolfi, "Engineering the evolution of self-organizing behaviors in swarm robotics: A case study," *Artificial Life*, vol. 17, pp. 183-202, 2011.

[95] M. Brambilla, C. Pinciroli, M. Birattari, and M. Dorigo, "Property-driven design for swarm robotics," in *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, 2012, pp. 139-146.

[96] M. Mesbahi, "On state-dependent dynamic graphs and their controllability properties," *Automatic Control, IEEE Transactions on*, vol. 50, pp. 387-392, 2005.

[97] K. Lerman, A. Martinoli, and A. Galstyan, "A Review of Probabilistic Macroscopic Models for Swarm Robotic Systems," in *Swarm Robotics*. vol. 3342, E. Şahin and W. Spears, Eds., ed: Springer Berlin Heidelberg, 2005, pp. 143-152.

[98] J. Barraquand and J. C. Latombe, "On nonholonomic mobile robots and optimal maneuvering," in *Intelligent Control, 1989. Proceedings., IEEE*

International Symposium on, 1989, pp. 340-347.

[99] F. Jean, "Complexity of nonholonomic motion planning," *International Journal of Control*, vol. 74, pp. 776-782, 2001.

[100] Y. Diaz-Mercado and M. Egerstedt, "Multi-robot mixing of nonholonomic mobile robots," in *Control Applications (CCA), 2014 IEEE Conference on*, 2014, pp. 524-529.

[101] L. E. Dubins, "On Curves of Minimal Length with a Constraint on Average Curvature, and with Prescribed Initial and Terminal Positions and Tangents," *American Journal of Mathematics*, vol. 79, pp. 497-516, 1957.

[102] B. Xuan-Nam, J. D. Boissonnat, P. Soueres, and J. P. Laumond, "Shortest path synthesis for Dubins non-holonomic robot," in *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, 1994, pp. 2-7 vol.1.

[103] C. Hanson, J. Richardson, and A. Girard, "Path planning of a Dubins vehicle for sequential target observation with ranged sensors," in *American Control Conference (ACC), 2011*, 2011, pp. 1698-1703.

[104] K. Savla, F. Bullo, and E. Frazzoli, "The coverage problem for loitering Dubins vehicles," in *Decision and Control, 2007 46th IEEE Conference on*, 2007, pp. 1398-1403.

[105] A. Balluchi, A. Bicchi, B. Piccoli, and P. Soueres, "Stability and robustness of optimal synthesis for route tracking by Dubins' vehicles," in *Decision and Control, 2000. Proceedings of the 39th IEEE Conference on*, 2000, pp. 581-586 vol.1.

[106] D. H. Jearl Walker, Robert Resnick *Principles of Physics*, 9th ed.: WILEY, 2010.

[107] Y. Petillot, I. Tena Ruiz, and D. M. Lane, "Underwater vehicle obstacle avoidance and path planning using a multi-beam forward looking sonar," *Oceanic Engineering, IEEE Journal of,* vol. 26, pp. 240-251, 2001.

[108] R. Mobus and U. Kolbe, "Multi-target multi-object tracking, sensor fusion of radar and infrared," in *Intelligent Vehicles Symposium, 2004 IEEE*, 2004, pp. 732-737.

[109] A. Flynn, "Combining sonar and infrared sensors for mobile robot navigation," *The International Journal of Robotics Research,* vol. 7, pp. 5-14, 1988.

[110] F. A. G. Windmeijer, *Goodness of Fit in Linear and Qualitative-Choice Models*: Amsterdam: Thesis Publishers, 1992.

[111] T. Guoqing, W. Zidong, and A. L. Williams, "On the construction of an optimal feedback control law for the shortest path problem for the Dubins car-like robot," in *System Theory, 1998. Proceedings of the Thirtieth Southeastern Symposium on*, 1998, pp. 280-284.

[112] *Monte Carlo Simulation*. Available: http://www.palisade.com/risk/monte_carlo_simulation.asp

[113] *Curve Fitting Toolbox*. Available: http://www.mathworks.com/products/curvefitting/

Publications

[1] Q. Zhang, G. Leng, and V. Govindaraju, "Duration of collision-free motion of unmanned vehicles in a confined area," *Robotica*, vol. FirstView, pp. 1-14, 2014.

[2] Q. Zhang, G. Leng, and V. Govindaraju, "Duration of Collision-free Motions of Multiple Dubins' Vehicles using Velocity Obstacles in Confined Area," *International Journal of Robotics and Automation* (under review).

[3] V. Govindaraju, G. Leng, and Z. Qian, "Multi-UAV Surveillance over Forested Regions," *Photogrammetric Engineering & Remote Sensing*, vol. 80, pp. 1129-1137, 2014.

[4] Z. Qian, G. Leng, and V. Govindaraju, "Multiple Unmanned Vehicle Operations in Confined Areas," in *Intelligent Systems (GCIS), 2013 Fourth Global Congress on*, 2013, pp. 341-345.

[5] V. Govindaraju, G. Leng, and Z. Qian, "Visibility-based UAV path planning for surveillance in cluttered environments," in *Safety, Security, and Rescue Robotics (SSRR), 2014 IEEE International Symposium on*, 2014, pp. 1-6.

Appendix I. MATLAB Code: the Time to First Collision for Vehicles with Zero Turn Radius

```
close all;
hold on;
NumberOfVehicles = 2; %the number of vehicles
drawflag=1;
factor=500; %Adjust this factor when scaling
DT=5e-3;%time step
R = 0.05; %Radius of sensing range
Rvehicle = 0.025; %Radius of the vehicle
sensingAngle = pi/3; %***********30~60
absVelocity = 1; %************ 0.2~2
angleMakeup = atan(Rvehicle/(2*R-Rvehicle));
Bound=[-1 1 -1 1]; %************unit square
BallColour=[[1 0 0];[1 0 0.5];[1 0.5 0];[0 1 0];[0 0 1];[1 1 0];[1 1 1];[0 0.3
0];[0 0 0];[0.65 0.65 0.65];[0 0.75 0.75];[0.3 0 0.6];[0.95 0.65 0.75]; [0.5 0.25
0];[0 0.2 0.4];[0.9 0.4 0.7];[0.4 0.2 0.3];[0.65 0.55 0.15];[0.25 0.35 0.25];[0.5
0 0]];
TableColour=[.4 .5 .8];
%Plot handle
axis(Bound);
```

```
set(gca,'Color',TableColour,'xcolor',TableColour,'ycolor',TableColour,'PlotBo
xAspectRatio', [1 abs((Bound(3)-Bound(4))/(Bound(2)-Bound(1)))
1],'xtick',[],'ytick',[])
%================================================
r = R*ones(NumberOfVehicles,1);
%Radii=========================================
for count1=1:NumberOfVehicles;
   for count2=1:count1,
      rmatrix(count2,count1)=2*R;
   end;
end;
rmatrix=rmatrix.*triu(abs(-1+eye(size(rmatrix))));
%Mass==========================================
X=[(Bound(2)-Bound(1) 2*R)*rand(NumberOfVehicles,1)+Bound(1)+R,
(Bound(4)-Bound(3) 2*R)*rand(NumberOfVehicles,1)+Bound(3)+R];
for j=1:NumberOfVehicles;
   for i=1:j;
      distmatrix(i,j)=sqrt((X(j,1)-X(i,1))^2+(X(j,2)-X(i,2))^2);%distance
      matrix
   end;
end;
%Initial Edge detection matrix
Botsmatrix=(distmatrix-rmatrix)+tril(abs(-
1+eye(size(distmatrix))))+eye(size(distmatrix));
while find(Botsmatrix<=0);%check whether the vehicles overlap
```

```
    X=[(Bound(2)-Bound(1)-2*R)*rand(NumberOfVehicles,1)+Bound(1)+R,
        (Bound(4)-Bound(3)-2*R)*rand(NumberOfVehicles,1)+Bound(3)+R];
    for j=1:NumberOfVehicles;
       for i=1:j;
          distmatrix(i,j)=sqrt((X(j,1)-X(i,1))^2+(X(j,2)-X(i,2))^2);
       end;
    end;
    Botsmatrix=(distmatrix-rmatrix)+tril(abs(-
1+eye(size(distmatrix))))+eye(size(distmatrix));
end
%Initial velocities
angleOfVelocity = 2*pi*rand(NumberOfVehicles,1);
V=absVelocity*[cos(angleOfVelocity)      sin(angleOfVelocity)];      %random
velocities of vehicles
%Plot starting positions
for k=1:NumberOfVehicles;
    h(k)      =      plot(X(k,1),X(k,2),'o','MarkerEdgeColor',BallColour(mod(k-
    1,length(BallColour))+1,:),'MarkerFaceColor',BallColour(mod(k-
    1,length(BallColour))+1,:),'MarkerSize',factor*r(k));
end


pause(1e-1);


%Loop
n = 0;
```

```
while drawflag==1;%no collision happens
    n = n + 1;
    for k=1:NumberOfVehicles;
        delete(h(k));%delete figure
    end
    %Edgedetecton positive
    for j=1:NumberOfVehicles;
        for i=1:2;
            d=X(j,i)+R-Bound(2*i);%distance between vehicles
            if d>=0
                dt=d/V(j,i);
                X(j,i)=X(j,i)-V(j,i)*dt;
                if i == 1
                    if V(j,2) >= 0
                        temp = V(j,1);%turn anti-clockwise
                        V(j,1) = -V(j,2);
                        V(j,2) = temp;
                    else
                        temp = V(j,1);%turn clockwise
                        V(j,1) = V(j,2);
                        V(j,2) = -temp;
                    end
                elseif i == 2
                    if V(j,1) >= 0
                        temp = V(j,1);%turn clockwise
```

```
                V(j,1) = V(j,2);
                V(j,2) = -temp;
            else
                temp = V(j,1);  %turn anti-clockwise
                V(j,1) = -V(j,2);
                V(j,2) = temp;
            end
        end
    end
end
%Edgedetecton negative================================
for j=1:NumberOfVehicles;
    for i=1:2;
        d=X(j,i)-R-Bound(2*i-1);
        if d<=0
            dt=d/V(j,i);
            X(j,i)=X(j,i)-V(j,i)*dt;
            if i == 1
                if V(j,2) >= 0
                    temp = V(j,1);      %turn clockwise
                    V(j,1) = V(j,2);
                    V(j,2) = -temp;
                else
                    temp = V(j,1);  %turn anti-clockwise
```

```
                V(j,1) = -V(j,2);
                V(j,2) = temp;
            end
        elseif i == 2
            if V(j,1) >= 0
                temp = V(j,1);  %turn anti-clockwise
                V(j,1) = -V(j,2);
                V(j,2) = temp;
            else
                temp = V(j,1);      %turn clockwise
                V(j,1) = V(j,2);
                V(j,2) = -temp;
            end
        end
    end
end
%Distance matrix
for j=1:NumberOfVehicles;
    for i=1:j;
        distmatrix(i,j)=sqrt((X(j,1)-X(i,1))^2+(X(j,2)-X(i,2))^2);
        if (i~=j) && (distmatrix(i,j) <= 2*Rvehicle)
            drawflag = 0;
        end;
    end;
```

```
end;
%Collision detection matrix
Botsmatrix=(distmatrix-rmatrix)+tril(abs(-
1+eye(size(distmatrix))))+eye(size(distmatrix));
%======================
if find(Botsmatrix<0); %if collision happens
   [I,J]=find(Botsmatrix<0);
   for i=1:length(I)
      vectorCenter = [X(J(i),1)-X(I(i),1) X(J(i),2)-X(I(i),2); X(I(i),1)-
      X(J(i),1) X(I(i),2)-X(J(i),2)];
      vectorVelocity = [V(I(i),1) V(I(i),2);V(J(i),1) V(J(i),2)];
      for k = 1:2;
         cosTheta(k)=dot(vectorCenter(k,:),vectorVelocity(k,:))/(norm(vector
         Velocity(k,:))*norm(vectorCenter(k,:)));
      end
      if cosTheta(1) > 0
         theta(1) = acos(cosTheta(1));
         if theta(1) <= sensingAngle + angleMakeup
            normdist=normr([X(I(i),1)-X(J(i),1) X(I(i),2)-X(J(i),2)]);
            vaA=(V(I(i),1)*normdist(1)+V(I(i),2)*normdist(2));
            vaB=(V(J(i),1)*normdist(1)+V(J(i),2)*normdist(2));
            dt=abs(2*R-distmatrix(I(i),J(i)))/(abs(vaA)+abs(vaB));
            X(I(i),:)=X(I(i),:)-V(I(i),:)*dt;
            X(J(i),:)=X(J(i),:)-V(J(i),:)*dt;
```

```
        checkClockwise(1)    =    vectorCenter(1,1)*vectorVelocity(1,2)-
        vectorCenter(1,2)*vectorVelocity(1,1);
        if checkClockwise(1) > 0%anti-clockwise
            temp = V(I(i),1);%turn anti-clockwise
            V(I(i),1) = -V(I(i),2);
            V(I(i),2) = temp;
        elseif checkClockwise(1) < 0%clockwise
            temp = V(I(i),1);%turn clockwise
            V(I(i),1) = V(I(i),2);
            V(I(i),2) = -temp;
        else                %in a line
            temp = V(I(i),1);%turn clockwise
            V(I(i),1) = V(I(i),2);
            V(I(i),2) = -temp;
        end
    end
end

if cosTheta(2) > 0
    theta(2) = acos(cosTheta(2));
    if theta(2) <= sensingAngle + angleMakeup
        if theta(1) > sensingAngle + angleMakeup
            normdist=normr([X(I(i),1)-X(J(i),1) X(I(i),2)-X(J(i),2)]);
            vaA=(V(I(i),1)*normdist(1)+V(I(i),2)*normdist(2));
            vaB=(V(J(i),1)*normdist(1)+V(J(i),2)*normdist(2));
```

```
                dt=abs(2*R-distmatrix(I(i),J(i)))/(abs(vaA)+abs(vaB));
                X(I(i),:)=X(I(i),:)-V(I(i),:)*dt;
                X(J(i),:)=X(J(i),:)-V(J(i),:)*dt;
            end
            checkClockwise(2)    =    vectorCenter(2,1)*vectorVelocity(2,2)-
            vectorCenter(2,2)*vectorVelocity(2,1);
            if checkClockwise(2) > 0%anti-clockwise
                temp = V(J(i),1);%turn anti-clockwise
                V(J(i),1) = -V(J(i),2);
                V(J(i),2) = temp;
            elseif checkClockwise(2) < 0%clockwise
                temp = V(J(i),1);%turn clockwise
                V(J(i),1) = V(J(i),2);
                V(J(i),2) = -temp;
            else              %in a line
                temp = V(J(i),1);%turn clockwise
                V(J(i),1) = V(J(i),2);
                V(J(i),2) = -temp;
            end
        end
    end
  end
end

%Propagation
```

```
    X=X+V*DT;
    %Plotting
    for k=1:NumberOfVehicles;
        h(k)    =    plot(X(k,1),X(k,2),'o',    'MarkerEdgeColor',BallColour(mod(k-
1,length(BallColour))+1,:),...
            'MarkerFaceColor',BallColour(mod(k-
1,length(BallColour))+1,:),'MarkerSize',factor*r(k));
    end
    drawnow;
end
T = n*DT
```

Appendix II. MATLAB Code: the Time to First Collision for Dubins' Vehicles

```
clear all;
clc;
% initial constants
NumberOfVehicles = 11; % the number of vehicles
R_ob = 0.25; Rab = 2*R_ob; % the equivalent radius of obstacle; Rab=2*R_obstacle
absVelocity = 1; a = 2.2; % speed and acceleration
R = absVelocity^2/a; % turning radius
omega = a/absVelocity; % angular speed
S0 = 2*R+Rab;%S0 = sqrt(Rab^2+2*Rab*R); % critical distance
factorS = 1; % factor
St = factorS*S0;%St = k*S0; %distance to turn
k1 = 2;
k2 = 1.2;
Bound = [-10 10 -10 10]; % [xl xr yl yr]
drawflag = 1;
turnflag = zeros(NumberOfVehicles,1);
TURNDIRECT = zeros(NumberOfVehicles,1);
DT=0.01;
n = zeros(NumberOfVehicles,1);
pos_cen = zeros(NumberOfVehicles,2);
```

```
pos_veh_rel0 = zeros(NumberOfVehicles,2);
Angleturn = zeros(NumberOfVehicles,1);
pos_cen_rel = zeros(NumberOfVehicles,2);
%%%%%%%%%%%%preparation of plot%%%%%%%%%%%
BallColour=[[1 0 0];[1 0 0.5];[1 0.5 0];[0 1 0];[0 0 1];[1 1 0];[1 1 0.2];...
    [0 0.3 0];[0 0 0];[0.65 0.65 0.65];[0 0.75 0.75];[0.3 0 0.6];[0.95 0.65 0.75];...
    [0.5 0.25 0];[0 0.2 0.4];[0.9 0.4 0.7];[0.4 0.2 0.3];[0.65 0.55 0.15];[0.25 0.35
      0.25];[0.5 0 0]];
TableColour=[.4 .5 .8];
factor=50; %Adjust this factor when scaling
%Plothandle
axis(Bound);
set(gca,'Color',TableColour,'xcolor',TableColour,'ycolor',
TableColour,'PlotBoxAspectRatio',[1      abs((Bound(3)-Bound(4))/(Bound(2)-
Bound(1))) 1],'xtick',[],'ytick',[])
%%%%%%%%%%%%%%%% initial %%%%%%%%%%%
% initial values

r = R_ob*ones(NumberOfVehicles,1); % radius matrix

for count1=1:NumberOfVehicles
    for count2=1:count1
        rmatrix(count2,count1)=2*R_ob;
    end
end
```

```
rmatrix=rmatrix.*triu(abs(-1+eye(size(rmatrix)))); %triu: Upper triangular part
of matrix


% X are the positions of the centers of vehicles
X=[(Bound(2)-Bound(1)-
2*R_ob)*rand(NumberOfVehicles,1)+Bound(1)+R_ob,  (Bound(4)-Bound(3)-
2*R_ob)*rand(NumberOfVehicles,1)+Bound(3)+R_ob];
% distmatrix includes the relative distances between vehicles
for j=1:NumberOfVehicles
   for i=1:j
      distmatrix(i,j)=sqrt((X(j,1)-X(i,1))^2+(X(j,2)-X(i,2))^2);
   end
end
%Initial Edgedetectionmatrix========
Botsmatrix=(distmatrix-rmatrix)+tril(abs(-
1+eye(size(distmatrix))))+eye(size(distmatrix));
while find(Botsmatrix<=0)
   X=[(Bound(2)-Bound(1)-
2*R_ob)*rand(NumberOfVehicles,1)+Bound(1)+R_ob,  (Bound(4)-Bound(3)-
2*R_ob)*rand(NumberOfVehicles,1)+Bound(3)+R_ob];
   for j=1:NumberOfVehicles;
      for i=1:j;
         distmatrix(i,j)=sqrt((X(j,1)-X(i,1))^2+(X(j,2)-X(i,2))^2);
      end
   end
```

```
    Botsmatrix=(distmatrix-rmatrix)+tril(abs(-
1+eye(size(distmatrix))))+eye(size(distmatrix));
end

%Initial velocities
angleOfVelocity = 2*pi*rand(NumberOfVehicles,1); %matrix:
NumberOfVehicle x 1
V = absVelocity*[cos(angleOfVelocity) sin(angleOfVelocity)]; %
NumberOfVehicle x 2

%%%%%%%%%%%% plot %%%%%%%%%%%%%
%Plot starting positions
for k=1:NumberOfVehicles;
    h(k) = plot(X(k,1),X(k,2),'o', 'MarkerEdgeColor',BallColour(mod(k-
1,length(BallColour))+1,:), 'MarkerFaceColor',BallColour(mod(k-
1,length(BallColour))+1,:),'MarkerSize',factor*r(k));
    axis(Bound);
    axis square;
    hold on
end

pause(1e-1);
N = round((pi/2)/(omega*DT)); %the number of steps to turn 90 degrees  (ceil)
num = 0;
%%%%%%%%loop begins: until collision happens%%%%%%%%
```

```
while drawflag == 1;
   %================== plot ====================
   num = num + 1;
   % delete plot
   for k=1:NumberOfVehicles;
      delete(h(k));
   end
   %============================================
   %check whether vehicle collides with boundary, if yes, let drawflag = 0
   for j=1:NumberOfVehicles
      for i=1:2  %Edgedetection positive
         d=X(j,i)+R_ob-Bound(2*i);
         if d >= 0
            drawflag = 0;
            fprintf('%dth collide with boundary +\n',j);
         end
      end
      for i=1:2  %Edgedetection negative
         d=X(j,i)-R_ob-Bound(2*i-1);
         if d <= 0
            drawflag = 0;
            fprintf('%dth collide with boundary -\n',j);
         end
      end
```

```
    %%%%%   Distancematrix: distance between vehicles  %%%
    for i=1:j
        distmatrix(i,j)=sqrt((X(j,1)-X(i,1))^2+(X(j,2)-X(i,2))^2);
        if (i~=j) && (distmatrix(i,j) <= Rab)
            drawflag = 0;           % if two vehicles collide, drawflag = 0
            fprintf('%dth and %dth collide\n',i,j);
        end
    end
end
%----------------------------------------------------------------------
angleOfVelocity = angle0to2pi(angleOfVelocity); % angle range [0,2*pi);
matrix: NumberOfVehicle x 1
%%%%%%%%%  for every vehicle, turnflag = 0, straight line  %%%%%
for j=1:NumberOfVehicles
    clear angleOfPos S S1 S2 boun_dis vector_rel sub_angle angle_col
    angleturn pos_veh_rel;
    if turnflag(j) == 0 %straight line
        corner = 1; %at the beginning, assume that the vehicle is not in the
        corner
        %====================================
        % check the direction of velocity, calculate the distance to the
        boundary
        if (angleOfVelocity(j) >= 0) && (angleOfVelocity(j) < pi/2) %0~pi/2
            angleOfPos = atan((Bound(4)-X(j,2))/(Bound(2)-X(j,1)));
            S1  =  abs((Bound(2)-X(j,1)));%/cos(angleOfVelocity);  %distance
```

```
    from vehicle to boundary along the direction of velocity
    S2 = abs((Bound(4)-X(j,2)));%/sin(angleOfVelocity);
    if angleOfPos > angleOfVelocity(j)
        S = S1;
        S_temp = S2;
        if (S2 < S1) %|| (S2 < (2*R+R_ob))
            corner = -1; %closer to the other boundary than that may be
            collide with
        end
    else
        S = S2;
        S_temp = S1;
        if (S1 < S2) %|| (S1 < (2*R+R_ob))
            corner = -1;
        end
    end
elseif (angleOfVelocity(j) >= pi/2) && (angleOfVelocity(j) <
pi) %pi/2~pi
    angleOfPos = atan((Bound(4)-X(j,2))/(Bound(1)-X(j,1)))+pi;
    S1 = abs((Bound(4)-X(j,2)));%/sin(angleOfVelocity); %distance
    from vehicle to boundary along the direction of velocity
    S2 = abs((Bound(1)-X(j,1)));%/cos(angleOfVelocity); % Bound(1)-
    X(1,1)<0; cos(angleOfVelocity)<0
    if angleOfPos > angleOfVelocity(j)
        S = S1;
```

```
        S_temp = S2;
        if (S2 < S1) %|| (S2 < (2*R+R_ob))
            corner = -1; %closer to the other boundary than that may be
            collide with
        end
    else
        S = S2;
        S_temp = S1;
        if (S1 < S2) %|| (S1 < (2*R+R_ob))
            corner = -1;
        end
    end
elseif (angleOfVelocity(j) >= pi) && (angleOfVelocity(j) <
3*pi/2) %pi~3*pi/2
    angleOfPos = atan((Bound(3)-X(j,2))/(Bound(1)-X(j,1)))+pi; %-/-
    S1 = abs((Bound(1)-X(j,1)));%/cos(angleOfVelocity); %distance
    from vehicle to boundary along the direction of velocity
    S2 = abs((Bound(3)-X(j,2)));%/sin(angleOfVelocity); % -/-
    if angleOfPos > angleOfVelocity(j)
        S = S1;
        S_temp = S2;
        if (S2 < S1) %|| (S2 < (2*R+R_ob))
            corner = -1; %closer to the other boundary than that may be
            collide with
        end
```

```
    else
        S = S2;
        S_temp = S1;
        if (S1 < S2) %|| (S1 < (2*R+R_ob))
            corner = -1;
        end
    end
elseif (angleOfVelocity(j) >= 3*pi/2) && (angleOfVelocity(j) < 2*pi) %pi~3*pi/2
    angleOfPos = atan((Bound(3)-X(j,2))/(Bound(2)-X(j,1)))+2*pi; %-/+
    S1 = abs((Bound(3)-X(j,2)));%/sin(angleOfVelocity); %distance from vehicle to boundary along the direction of velocity
    S2 = abs((Bound(2)-X(j,1)));%/cos(angleOfVelocity); % +/+
    if angleOfPos > angleOfVelocity(j)
        S = S1;
        S_temp = S2;
        if (S2 < S1) %|| ((S2 < (2*R+R_ob))&&(S2 > S1))
            corner = -1; %closer to the other boundary than that may be collide with
        end
    else
        S = S2;
        S_temp = S1;
        if (S1 < S2) %|| ((S1 < (2*R+R_ob))&&(S1 > S2))
```

```
            corner = -1;
        end
    end
end
%================================================
% if the distance to the boundary S<(R+R_ob) (not corner) or
S<(2R+R_ob) (corner), turn 90 degrees
if corner == -1
    boun_dis = k1*R+R_ob; %k1=2
else
    boun_dis = k2*R+R_ob; %k2=1.2
end
if (S < boun_dis)||(S_temp <= k1*R+R_ob)
    turnflag(j) = 1;
    if angleOfPos > angleOfVelocity(j)
        TURNDIRECT(j) = 1*corner; % turn anti-clockwise; if in the
         corner, clockwise
    else
        TURNDIRECT(j) = -1*corner; % turn clockwise; if in the corner,
        anti-clockwise
    end
else
    for i=1:j
        if turnflag(i) == 0
            temp_Xj=X(j,:)+V(j,:)*DT; % j one step ahead to check
```

```
obstacle velocity with i (because i<j, i has one step forward
than j)
distmatrix_temp = sqrt((temp_Xj(1)-X(i,1))^2+(temp_Xj(2)-
X(i,2))^2);
vector_rel = X(i,:)-temp_Xj; %relative vector between vehicles
sub_angle= acos(dot(vector_rel,V(j,:)-
V(i,:))/(norm(vector_rel)*norm(V(j,:)-V(i,:)))); %angle
between velocity and relative vector
angle_col = asin(Rab/distmatrix_temp); % the angle between
OiOj(relative vector) and tangent line
if (i~=j)&&(sub_angle < angle_col) && (distmatrix(i,j) < St)
   fprintf('%d and %d are avoiding,
   distance=%f\n',i,j,distmatrix(i,j));
   turnflag(i) = 1;
   turnflag(j) = 1;
  TURNDIRECT(j) = turndirec(vector_rel,V(j,:)-
   V(i,:)); %judge the direction from relative vector(obstacle -
   reference) to velocity
   TURNDIRECT(i) = TURNDIRECT(j);
   % set parameters for i while turning
 pos_cen_rel(i,:)=(R/absVelocity)*after_rot(TURNDIRECT(i)
*(pi/2),V(i,:));
   pos_cen(i,:) = X(i,:) + pos_cen_rel(i,:);
   n(i) = N;
   pos_veh_rel0(i,:) = -pos_cen_rel(i,:);
```

```
                Angleturn(i) = TURNDIRECT(i)*omega*DT;
                break;
            end
        end
    end
end

%====================================

if turnflag(j) == 1 % if need to turn,set initial values for turning
    %the coordinate of the center of circle relative to the position of
    vehicle: pos_cen_rel
    pos_cen_rel(j,:)=(R/absVelocity)*after_rot(TURNDIRECT(j)*(pi/2)
    ,V(j,:));
    %the coordinate of the center of circle
    pos_cen(j,:) = X(j,:) + pos_cen_rel(j,:);
    n(j) = N; %let the counter equals the number of steps
    pos_veh_rel0(j,:) = -pos_cen_rel(j,:); %coordinate of vehicle relative
    to center of circle
    Angleturn(j) = TURNDIRECT(j)*omega*DT;
elseif turnflag(j) == 0 % after checking, if still no need to turn
    X(j,:)=X(j,:)+V(j,:)*DT;
end
end % end of turnflag == 0
%%%%%%%turnflag = 1, turning %%%%%%%
```

```
    % if the vehicle needs to turn, turnflag = 1, and turn until 90 degrees
    if turnflag(j) == 1
        angleturn = Angleturn(j)*(N-n(j)+1);
        pos_veh_rel = after_rot(angleturn,pos_veh_rel0(j,:)); %coordinate of
         vehicle relative to center (turning)
        X(j,:) = pos_veh_rel + pos_cen(j,:); %coordinate of vehicle
        if n(j) == 1 %at last, calculate velocity
            fprintf('%d finish turning\n',j);
            turnflag(j) = 0;
            V(j,:)=(absVelocity/R)*after_rot(TURNDIRECT(j)*(pi/2),pos_veh_
                    rel); %derive the velocity by turning pos_veh_rel 90 degrees
                    angleOfVelocity(j)=(V(j,2)/abs(V(j,2)))*acos(dot([1,0],V(j,:))
                    /(norm([1,0])*norm(V(j,:)))); %subtended angle with (1,0)
        end
        n(j) = n(j)-1;
    end
end
%Plotting=============================================
for k=1:NumberOfVehicles;
    h(k)    =    plot(X(k,1),X(k,2),'o','MarkerEdgeColor',BallColour(mod(k-
                  1,length(BallColour))+1,:),'MarkerFaceColor',BallColour(
                  mod(k-1,length(BallColour))+1,:),'MarkerSize',factor*r(k));
end
drawnow;
pause(1e-2);
```

```
end

t = (num-1)*DT
```

Appendix III. Simulation Environment

A. Monte Carlo simulation

Monte Carlo simulation [112] is the name of a broad class of simulation methods that solve for numerical solutions to problems by repeating random samplings, which can help find the impact of risk and uncertainty in the models. This technique was first applied in studying the atomic bomb. Monte Carlo simulation is commonly used in obtaining numerical solutions when the problem is too complicated for an analytical solution, or when some theory needs to be verified using simulation, as the case in this study. Possible results can be obtained using random numbers, so the probable distribution of the results is revealed. Besides, this technique can be used to estimate the range of values, so that we can know possible outcomes in the future from the mean values and standard deviations of the results. Moreover, the probability distribution of the results that is affected by variables can also be found, by changing the range of the values of variables. Although only random distribution results are calculated, the trend of results and the mean values of outcomes are valuable for studies by repeating the simulation over and over again, for example in Figure A.1 the simulation points can still reveal the trend of the theoretical curve despite of discretization. Computer makes it possible to do quantitative numerical analysis. Therefore, Monte Carlo simulation has been used in a variety of fields, such as mathematics, engineering, insurance. It is especially useful in sensitivity analysis and quantitative probabilistic analysis in process design.

In the simulation, the ranges of random variables will be given, and the random values are generated every time Monte Carlo simulation runs, while the model is calculated hundreds or thousands of times typically. The result will rely on the random values generated. The large amount of results after simulation can be analyzed to describe the performance of the model and predict the results in the future. This kind of simulation can only predict the probability of the results rather than give a certain value.

The studies in this thesis adopt Monte Carlo simulation to verify the theories that are developed. The initial parameters are generated randomly, and the operations of vehicles are fully simulated based on the scenario we set. Finally, the time to first collision is collected as result.

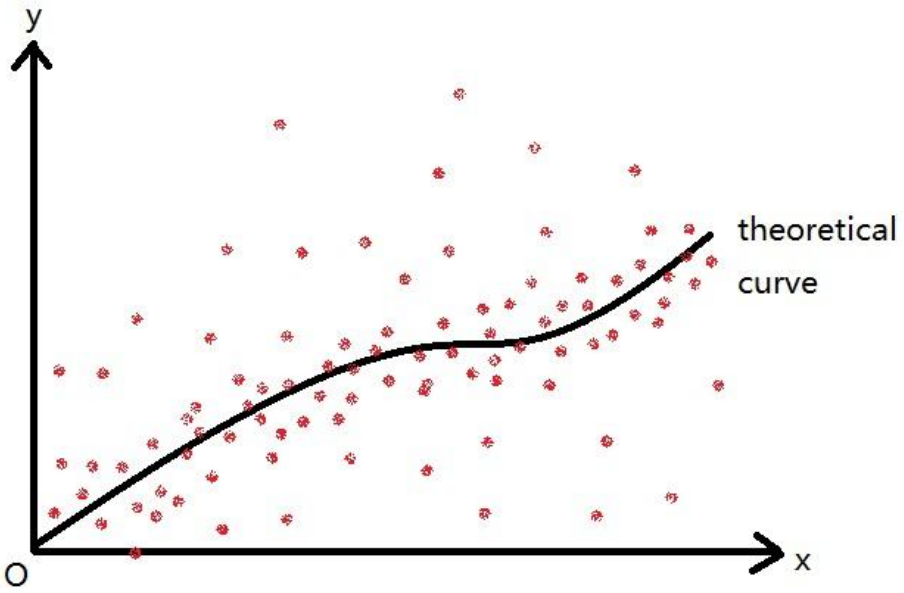


Figure A.1 Diagrammatic sketch of Monte Carlo simulation. The red dots are simulation points and the black curve is theoretical curve.

B. Curve Fitting Toolbox

A large amount of data will be obtained after Monte Carlo simulation, and the results are analyzed by Curve Fitting Toolbox in MATLAB [113]. The Curve Fitting Toolbox is a kind of graphical user interfaces (GUIs) for fitting curves or surfaces to data. Exploratory data analysis and data process can be performed by the toolbox. This toolbox can preprocess data like sectioning or smoothing. It also allows us to define our own custom equations, other than linear and nonlinear models that are provided in the toolbox. There are many library equations in the toolbox, such as polynomials, exponentials, rationals, sums of Gaussians, so parametric fit can be performed. Nonparametric fit can also be done by smoothing spline or interpolants. All kinds of least squares are supplied in the toolbox. The quality of fit can be dramatically improved through the library. Besides, many statistic methods can help determine the goodness of fit. The distinguishing feature of this toolbox is the graphical environment, which allows us to fit and analyze data sets visually and explicitly. However, the functions in Curve Fitting Toolbox can still be used in MATLAB command line environment. The Curve Fitting Tool and the command-line environment can not complete a curve fitting task at the same time cooperatively. After fitting is done, many post-processing methods for plotting can be applied, for example interpolation, and extrapolation. In addition, confidence range can also be estimated, and integrals and derivatives can be calculated.

The interface of Curve Fitting Tool is shown in Figure B.1. First, the data that exist in the MATLAB workspace should be imported to Curve Fitting

Tool. Then, the form of fitting curve is selected by us, and the data are fitted to the curve by examining the fit results graphically and numerically. The residuals can also be shown in the figure. Finally, the constants will be obtained from the box.

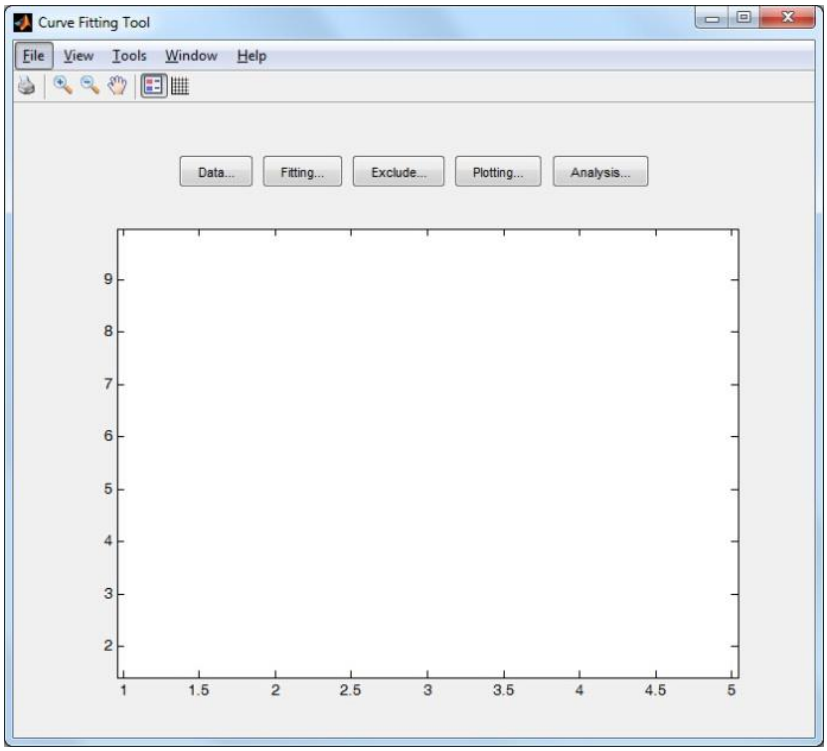


Figure B.1 The interface of Curve Fitting Tool in MATLAB.